

**DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL  
FACULTAD DE INFORMÁTICA**

**MODELOS FLEXIBLES DE  
CONOCIMIENTO ESTRUCTURADO  
PARA SOPORTE DE  
INTERFACES INTELIGENTES  
PARA AYUDA A LA DECISIÓN**

**AUTORA: JOSEFA ZULEIDE HERNÁNDEZ DIEGO  
DIRECTOR: JOSÉ CUENA BARTOLOMÉ  
CODIRECTOR: MARTÍN MOLINA GONZÁLEZ**

**1998**



Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad  
Politécnica de Madrid, el día 23 de septiembre de 1998.

- Presidente: D. Manuel Hermenegildo Salinas
- Vocal: D<sup>a</sup>. María Felisa Verdejo Maíllo
- Vocal: D. Julio Abascal González
- Vocal: D. Alfredo Fernández Valmayor
- Secretario: D<sup>a</sup>. Ana García Serrano

Realizado el acto de defensa y lectura de la Tesis en Madrid el día ..... de  
..... de 1998

Calificación: .....

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO





*A mis padres*



# Agradecimientos

Esta tesis es el resultado de varios años de trabajo e investigación a lo largo de los cuales he conocido a un gran número de personas que han influido de una u otra manera en el desarrollo de este trabajo y a los cuales estoy agradecida. Sin embargo, me gustaría destacar entre ellas...

A José Cuenca, por darme la oportunidad de disfrutar trabajando con él desde hace varios años, por su dedicación y paciencia, y por darme mucho más que conocimientos de Inteligencia Artificial.

A Martín Molina, por su apoyo en la realización de este trabajo y las divertidas discusiones sobre las posibilidades de los métodos de resolución de problemas.

A Ana García Serrano, por escuchar y revisar con interés las diferentes versiones de esta tesis que le he ido presentando, por sus buenos consejos y los cafés matutinos, por su amistad.

A Reyes Riera, por sus constantes ánimos, y por soportar con infinita paciencia mis periódicas ocupaciones de su Macintosh.

A todos mis amigos y compañeros del ISYS, los actuales y los que ya se han ido. Con ellos he disfrutado del mejor entorno de trabajo posible, y también de las mejores cañas, cenas y fiestas. Compartir con ellos estos años lo ha hecho todo más fácil, y este trabajo le debe más a ellos de lo que se imaginan.

Y sobre todo, a mis padres y hermanas, por su constante y total confianza y apoyo. Ellos han sido, en muchos momentos, el motor que me ha impulsado a conseguir que este volumen acabase entre sus manos.

Madrid, Noviembre 1998



## **Resumen**

En los últimos años, el desarrollo de la tecnología telemática ha introducido cambios significativos en los escenarios de operación en tiempo real, que han hecho necesaria la presencia de sistemas informáticos que facilitasen a los operadores el aprovechamiento de la gran cantidad de información recibida y proporcionasen soporte en el proceso de toma de decisiones. El éxito de esta cooperación para la realización de tareas depende de la capacidad del sistema para comunicarse adecuadamente con los usuarios.

Esta tesis aborda el problema de enriquecer el nivel de interacción entre sistemas y usuarios mediante la incorporación de una inteligencia capaz de explotar las capacidades de resolución de problemas del sistema en la dirección marcada por las características de un diálogo. Esta inteligencia surge de la integración de ideas provenientes de los Sistemas Inteligentes y de los Interfaces de Usuario Inteligentes, y toma forma en una arquitectura reflexiva, llamada ROHCI, en la que se distinguen los siguientes elementos:

- Un modelo de interacción estructurado por clases de preguntas que define los elementos en torno a los que se desarrollará el proceso comunicativo.
- Un conjunto de métodos de resolución de problemas, soportados por conocimiento del dominio y organizados como un modelo estructurado de conocimiento, con los que generar la información solicitada por el usuario en sus consultas.
- Un razonador metanivel, donde reside la capacidad reflexiva del sistema, que decide dinámicamente el tipo de respuesta que debe proporcionarse al usuario y los métodos de resolución de problemas que deben intervenir para obtenerla, en función de las características del usuario y el contexto del diálogo.
- Una memoria del diálogo, en donde se mantiene de forma consistente el conjunto de creencias comunes al sistema y el usuario generadas durante el diálogo y que asegura la coherencia de la interacción usuario-sistema en el tiempo.

La viabilidad de la arquitectura ROHCI se muestra con su aplicación en un problema de dimensión real, diseñando y construyendo un sistema de ayuda a la decisión en la gestión del transporte público de la ciudad de Turín.



## **Abstract**

In the last years, the evolution of the telematics technology introduced significant changes in the real time operation scenarios that made necessary the development of advanced information systems. These systems should facilitate the operators the management of large amounts of information and support these operators in the decision making process. The success of this cooperative task performance lies on the capacity of the system to adequately communicate the information to the users.

This thesis aims to improve the user-system interaction level by means of an intelligence capable to exploit the problem solving capabilities of a system according to the characteristics of a dialogue. This intelligence results from the integration of ideas taken from the Intelligent Systems and Intelligent User Interfaces research areas, and it is supported by a reflective architecture, named ROHCI, where the following elements can be distinguished:

- An interaction model structured in classes of questions that establishes the elements to be used in a communicative process.
- A collection of problem solving methods, supported by domain knowledge and organized in a structured knowledge model, that are used to generate the information required by the users.
- A metalevel reasoner, supporting the reflective capability of the system, which dynamically decides the type of answer to be provided to the user and the problem solving methods that should generate this answer, according to the characteristics of the user and the context of the dialogue.
- A memory of the dialogue that includes a consistent set of assumptions generated along the dialogue that are common to the user and the system. This memory supports the coherence of the user-system interaction along the time.

The feasibility of the ROHCI architecture is illustrated with its application to a real world problem. It was used to design and implement a decision support system to manage the public transport in the city of Turin.

# Indice

---

## INTRODUCCIÓN

- 1. INTRODUCCION.....3
  - 1.1 Motivación.....3
  - 1.2 Objetivos .....6
  - 1.3 Organización .....8

## PARTE I: ANTECEDENTES

- 2. INTERACCIÓN USUARIO-SISTEMA COMO RESULTADO DE LA ESTRUCTURACIÓN DEL CONOCIMIENTO.....13
  - 2.1 La Modelización del Conocimiento .....14
    - 2.1.1 El Nivel de Conocimiento de Newell .....14
    - 2.1.2 El modelo de Tareas Genéricas.....16
    - 2.1.3 Los métodos "Role Limiting" .....19
    - 2.1.4 Los niveles de conocimiento de KADS.....21
    - 2.1.5 El modelo de Componentes de la Experiencia .....23
    - 2.1.6 El modelo de Unidades Cognitivas.....24
  - 2.2 Modelos Estáticos de Conocimiento.....27
    - 2.2.1 KREST.....28
    - 2.2.2 PROTÉGÉ-II.....30
    - 2.2.3 KSM.....32
    - 2.2.4 Instrumentaciones de KADS.....35
  - 2.3 Modelos Dinámicos de Conocimiento .....39
    - 2.3.1 TIPS.....40
    - 2.3.2 MODEL-K .....43
    - 2.3.3 TroTelC.....45

2.4 Conclusiones .....	47
3. INTERACCIÓN USUARIO-SISTEMA COMO RESULTADO DE UNA PRESENTACIÓN ELABORADA .....	49
3.1 Presentaciones de la Primera Generación.....	50
3.1.1 Los patrones de la estructura de un discurso .....	51
3.1.2 La teoría de Estructuras Retóricas.....	53
3.2 Presentaciones Inteligentes en Lenguaje Natural.....	55
3.2.1 El modelo de planificación de Moore y Paris .....	56
3.2.2 La teoría de actos comunicativos de Maybury.....	59
3.3 Presentaciones Inteligentes Multimedia .....	63
3.3.1 Las acciones retóricas ampliadas de Maybury .....	63
3.3.2 WIP.....	65
3.3.3 COMET.....	69
3.4 Modelos de Usuario .....	73
3.4.1 Problemática de la modelización de usuarios .....	75
3.4.2 Representación de modelos de usuario.....	76
3.5 Diseño de Interfaces de Usuario Basado en Modelos .....	84
3.6 Conclusiones .....	87
4. ANÁLISIS.....	89
4.1 Resumen de las propuestas.....	89
4.2 Problemas pendientes.....	95

## **PARTE II: PROPUESTA**

5. INTERACCIÓN USUARIO-SISTEMA AVANZADA .....	101
5.1 Planteamiento General.....	101
5.2 Características del Modelo de Interacción Usuario-Sistema.....	104
5.2.1 Cualidades comunicativas del diálogo.....	104
5.2.2 Comportamiento del sistema .....	114



6. MODELO DE CONOCIMIENTO PARA SOPORTAR INTERFACES INTELIGENTES .....	117
6.1 Arquitectura General .....	117
6.2 El Modelo de Interacción.....	123
6.3 El Espacio de Resolvedores de Problemas.....	125
6.3.1 Elementos.....	125
6.3.2 Estructuración .....	126
6.4 El Razonador Metanivel .....	129
6.4.1 Descripción general.....	130
6.4.2 Modelo de escenarios de interacción .....	132
6.4.2.1 Atributos de interacción.....	133
6.4.2.2 Representación del conocimiento .....	138
6.4.2.3 Caracterización del escenario de interacción .....	141
6.4.3 Modelo reflexivo de resolución de problemas .....	142
6.4.3.1 Representación del conocimiento .....	143
6.4.3.2 Diseño de un modelo de razonamiento.....	149
6.5 El Contexto de la Interacción .....	150
6.5.1 Representación del conocimiento.....	153
6.5.2 Gestión de la memoria .....	157
6.5.2.1 Adición de información.....	158
6.5.2.2 Acceso a información.....	159
6.5.2.3 Revisión de información .....	161
6.5.2.4 Eliminación de información.....	162
6.6 Conclusiones .....	163

## **Parte III: APLICACIÓN**

7. GESTIÓN DE TRANSPORTE PÚBLICO EN TIEMPO REAL.....	169
7.1 Sistemas de Ayuda a la Decisión en Tiempo Real.....	169
7.2 El Problema de la Gestión del Transporte Público.....	172
7.2.1 Terminología de gestión de transporte público .....	175

7.2.2 Clases de problemas.....	176
7.2.3 Acciones de control.....	177
7.3 El Modelo de Interacción Usuario-Sistema .....	181
7.3.1 Elementos del interfaz de usuario.....	185
7.4 El Espacio de Resolvedores de Problemas.....	188
7.4.1 Resolución de problemas clasificativos .....	189
7.4.1.1 Revisión de datos.....	190
7.4.1.2 Clasificación de estados.....	191
7.4.2 Resolución de problemas predictivos.....	194
7.4.3 Resolución de problemas de planificación.....	196
7.5 El Razonador Metanivel .....	199
7.5.1 Modelo de escenarios de interacción .....	199
7.5.2 Modelo reflexivo de resolución de problemas .....	204
7.6 Conclusiones .....	212
8. IMPLEMENTACIÓN SOBRE KSM.....	215
8.1 Modelo Genérico de Gestión de Transporte Público .....	216
8.2 Modelo de Tareas y Métodos.....	221
8.3 Modelo del Dominio de la Red de Autobuses de Turín.....	225
8.4 Ejemplo de Operación.....	230
8.5 Conclusiones .....	235

## CONCLUSIONES

9. EVALUACIÓN.....	239
9.1 Evaluación desde la Perspectiva de los Sistemas Inteligentes.....	239
9.2 Evaluación desde la Perspectiva de los Interfaces de Usuario Inteligentes .....	251

10. CONCLUSIONES.....255

10.1 Contribuciones.....255

10.2 Limitaciones.....260

10.3 Trabajos Futuros.....261

11. REFERENCIAS.....263

**ANEXO**

A. Gestión del Diálogo Multimedia en el Proyecto FLUIDS .....283

A.1 Introducción.....283

A.2 Interacción con el Modelo de Resolución de Problemas.....284

A.3 Interacción Multimedia Basada en Planes.....286

A.4 Proceso de Generación de Presentaciones.....288



# INTRODUCCIÓN



# 1. Introducción

---

Hoy día el acerbo de tecnologías de Inteligencia Artificial hace posible la integración de grupos de ellas al servicio de una clase de problemas cuya configuración constituye realmente una contribución al avance de la informática, ya que definen en sí formas de productos originales e innovadores.

Esta tesis es un ejemplo de este tipo de enfoque. En ella se ha identificado un conjunto de cuestiones abiertas en el área de la interacción entre sistemas informáticos y personas, y se ha propuesto una arquitectura apoyada en técnicas y módulos preexistentes en Inteligencia Artificial cuya integración supera algunas de las deficiencias observadas en el estado del arte actual. De acuerdo con este concepto, la metodología seguida ha sido estudiar las soluciones disponibles para comunicar adecuadamente personas y sistemas, identificando sus limitaciones actuales, y proponer una arquitectura cuyo uso reduce y clarifica algunas de estas limitaciones. Esta arquitectura se valida demostrándola con el tratamiento de un caso práctico tomado del proyecto FLUIDS [FLUIDS, 96].

## 1.1 Motivación

Existen varias definiciones sobre lo que constituye la nueva disciplina llamada Interacción Persona-Computador (*Human-Computer Interaction*, HCI). Una de las más comúnmente aceptadas es la caracterización formulada por el Curriculum Development Group del ACM Special Interest Group en Human-Computer Interaction (SIGCHI): *"una disciplina dedicada al diseño, evaluación e implementación de sistemas informáticos interactivos a ser usados por personas, y al estudio de los aspectos fundamentales que los rodean"* [Hewett et al., 92], [Abascal, 96]. En términos más informales se puede definir como una disciplina centrada en todos los aspectos que rodean la relación de una persona con un computador, ya sean estos físicos, psicológicos o teóricos.

La presencia, cada vez más importante, de los ordenadores en los centros de trabajo debida al desarrollo de las tecnologías de la información y a su capacidad para facilitar, y en ocasiones asumir, tareas tradicionalmente desempeñadas por personas, ha hecho necesario el estudio de todos los aspectos relevantes de la interacción de un usuario con su computador para asegurar un buen rendimiento. Esta cuestión ha pasado a ser especialmente relevante desde el momento en que la responsabilidad sobre la tarea realizada ha dejado de residir en el operador o el sistema para pasar a ser compartida por ambos. Es en este sentido de realización cooperativa de tareas en el que alcanzar un grado de comunicación adecuado entre un sistema y su usuario se convierte en un aspecto fundamental, y es por ello que el centro de las investigaciones en HCI es el usuario y la mejora de la capacidad comunicativa de las aplicaciones hacia sus usuarios.

Pero además, esta necesidad de construir sistemas con los que los usuarios puedan alcanzar un nivel de interacción adecuado ha dado paso al concepto de usabilidad y a la definición de metodologías de diseño de sistemas específicamente centradas en los usuarios [Gould, Lewis]. La usabilidad de un producto, según la norma ISO 9241, puede entenderse como una medida de las posibilidades que ofrece el producto a un grupo específico de usuarios para llevar a cabo sus objetivos con efectividad, eficiencia y satisfacción. Las metodologías de diseño centradas en usuarios persiguen estos objetivos mediante la incorporación de la perspectiva de los usuarios en el proceso de desarrollo del software. Para ello, se hace participar activamente a los usuarios finales de las aplicaciones en las etapas de diseño y evaluación de diferentes prototipos desarrollados a partir de las críticas que estos mismos usuarios hayan hecho a versiones anteriores, haciendo que tras sucesivos refinamientos se alcance un producto final que satisfaga tanto los requisitos funcionales y técnicos previstos como los propios requisitos de usabilidad [Daly-Jones et al., 97].

Ahora, volviendo a la necesidad de mejorar el nivel de comunicación entre sistemas y usuarios para poder soportar una realización eficiente de tareas de forma cooperativa, se plantea el problema de que dada la gran disparidad a priori de los modelos conceptuales manejados por un operador humano y un sistema para resolver un problema, si éste ha de ser resuelto de forma conjunta o cooperativa por ambos es necesario disponer de un intérprete que salve esa distancia y reduzca diferencias. Este ha sido el objetivo que ha guiado la



investigación en una de las áreas de estudio del HCI, la de los denominados *interfaces de usuario inteligentes*, centrada en el desarrollo de las capacidades expresivas de los interfaces para poder elaborar presentaciones adaptadas a las características del usuario.

Sin embargo, el comportamiento sofisticado o aparentemente inteligente de un sistema no hace que el sistema sea efectivamente inteligente. La calificación de un sistema como inteligente ha de estar basada en consideraciones más allá de las brillantes presentaciones elaboradas por su interfaz: éste ha de ser capaz de proporcionar explicaciones adecuadas sobre la forma en que razona y el conocimiento en el que fundamenta sus conclusiones. Es decir, al mismo tiempo que se mejora la inteligencia del vehículo de transmisión de información, es necesario enriquecer la información en sí misma aumentando las funcionalidades del modelo de resolución de problemas.

En los últimos años, las investigaciones en este área del HCI han girado en torno a la potenciación de las capacidades expresivas de los interfaces en términos de soporte de diálogos y uso de medios avanzados. En esta línea, las propuestas más avanzadas son las de los interfaces basados en el conocimiento, que generan presentaciones multimedia coordinadas de información extraída de bases de datos o proporcionada por un módulo de resolución de problemas. De esta manera se eleva el grado de interacción usuario-sistema al tener disponibles simultáneamente varios canales de transmisión de información.

Sin embargo, estas presentaciones avanzadas no habrían pasado de ser malabarismos técnicos sin el soporte de modelos de interacción en el que un componente fundamental han sido los modelos de usuario, en los que se incluyen los principales aspectos del modelo conceptual del usuario para guiar la especificación del aspecto visual de la interfaz. De esta manera, en cada etapa del diálogo usuario-sistema, se decide qué decir y cómo decirlo en la forma más ventajosa para el usuario.

Para asegurar este tipo de funcionalidad, los modelos de interfaces han ido incorporando progresivamente elementos propios de las arquitecturas basadas en el conocimiento que facilitaran el acceso y mantenimiento de los distintos modelos componentes (diálogo, presentación, tarea, usuario) a la vez que proporcionaban al usuario un nivel de servicio adecuado.



Sin embargo, un modelo de interacción usuario-sistema adecuado no debe quedarse en la elaboración de presentaciones brillantes. En el caso de sistemas interactivos en los que el usuario se apoya en información generada por el sistema para tomar decisiones, éstos deben ser capaces de ganarse la confianza de su usuario haciendo visible su forma de razonar; es decir, ha de conseguirse que el usuario crea en el modelo de resolución de problemas empleado por el sistema.

Para alcanzar este objetivo, es necesario que el usuario sea capaz de saber cómo y por qué el sistema ha generado los resultados que presenta. Este tipo de servicio requiere la aplicación de arquitecturas basadas en el conocimiento, no sólo para los modelos de presentación, sino también para el modelo de resolución de problemas. Por tanto, el diseño de una aplicación, vista como un par usuario-sistema, debería contemplar la definición de modelos basados en el conocimiento de todos los procesos de comprensión aplicados por el sistema.

De esta manera puede darse un paso adicional en el acercamiento de los modelos conceptuales de un usuario y un sistema para conseguir que la aplicación parezca algo más que una herramienta sofisticada de adquisición y gestión de información: un entorno para resolución de problemas completo, capaz de justificar sus conclusiones y ofrecer soluciones alternativas mediante el desarrollo de una sucesión de preguntas y respuestas adaptadas a las características y necesidades del usuario, y a la medida del problema que se resuelve en cada momento.

Esta tesis aborda el problema de enriquecer el nivel de interacción entre sistemas y usuarios mediante la incorporación de una inteligencia capaz de explotar las capacidades de resolución de problemas de un sistema en la dirección marcada por las características de un diálogo con un usuario.

## 1.2 Objetivos

El objetivo general de esta tesis es proporcionar un modelo de organización de conocimiento con el que puedan soportarse diálogos entre sistemas y usuarios, definidos como secuencias de preguntas y respuestas desarrolladas en torno a los objetivos de los usuarios. Para ello debe identificarse la finalidad que persigue el usuario al plantear consultas al sistema, buscando dinámicamente

el modo de satisfacerla en función de las capacidades de resolución de problemas del sistema.

Entonces, el objetivo de esta tesis se centra en proponer un conjunto de conceptos que sirvan de guía al diseño y desarrollo de sistemas de estas características, para lo cual deben alcanzarse los siguientes objetivos parciales:

(1) Analizar y evaluar las técnicas empleadas actualmente para conseguir un nivel de comunicación satisfactorio entre un sistema y sus usuarios. Estas técnicas se analizan desde dos perspectivas bien diferenciadas:

- la que presentan los sistemas abiertos, basados en organizaciones estructuradas de conocimiento, que buscan el acercamiento de los usuarios mediante representaciones del modelo de comprensión de un dominio problema cercanas a la forma en que estos usuarios razonan en ese dominio;
- la que ofrecen las técnicas de desarrollo de interfaces de usuario tendentes a enriquecer la capacidad expresiva de los sistemas para asegurar el éxito de la comunicación vía el uso de múltiples medios.

(2) Proponer un modelo flexible de organización de conocimiento que soporte el mantenimiento de diálogos usuario-sistema articulados en torno a las necesidades comunicativas del tipo de conversación usuario-sistema que se plantee, proporcionando por tanto elementos enriquecedores no sólo para la generación de presentaciones adaptadas a los usuarios sino también para generar un contexto de comprensión mutua usuario-sistema.

(3) Mostrar la viabilidad computacional del modelo de interacción anterior mediante el diseño de un entorno de desarrollo de sistemas con capacidad para decidir dinámicamente la forma en que deben interactuar con sus usuarios, explotando en esta dirección sus conocimientos sobre resolución de problemas.



- (4) Mostrar la validez del entorno anterior mediante su aplicación al desarrollo de sistemas de ayuda a la decisión en el dominio de gestión del transporte público.

Este trabajo define una nueva etapa en la línea de investigación en Ingeniería del Conocimiento que dio lugar a la tesis de [Molina, 93], orientada al desarrollo de sistemas abiertos basados en conocimiento estructurado que faciliten la comprensión, el mantenimiento y la reusabilidad de las aplicaciones. Surge asimismo, a partir de la experiencia adquirida en la especificación de los requisitos y funcionalidades deseables de los sistemas de ayuda a la decisión en tiempo real [Cuenca, Hernández, 97], obtenidos de sistemas como CYRAH [Cuenca et al., 89], SIRAH [Alonso et al., 90], KITS [Cuenca et al., 94], TRYs [Cuenca et al., 95, 96a, 96b], ARTEMIS [Serrano, 96]. Con esta propuesta se mejora la usabilidad de los mismos al dotarlos de modelos de comunicación usuario-sistema flexibles y adaptables a las características de los interlocutores, como se muestra en el prototipo desarrollado para la gestión en tiempo real de transporte público.

## 1.3 Organización

La línea argumental de la memoria de la tesis se basa en el análisis de las soluciones alternativas disponibles para mejorar el nivel de interacción entre personas y sistemas desde dos perspectivas: la que ofrecen los modelos estructurados de conocimiento y la que proporcionan los interfaces de usuario avanzados. Las deficiencias observadas en estas soluciones son el punto de partida a una nueva solución que busca la integración de los aspectos positivos de ambas líneas de investigación y la superación de algunas de sus limitaciones.

La memoria está estructurada en cuatro partes:

- ⇒ En la Parte I se describe el marco teórico sobre el que se asienta esta tesis. El capítulo 2 está dedicado a la descripción de las diferentes propuestas de modelización y estructuración del conocimiento que han hecho posible mejorar la comunicación entre usuarios y sistemas mediante la construcción de sistemas abiertos, accesibles a los usuarios y capaces de justificar sus conclusiones. El capítulo 3 presenta la

evolución de diferentes elementos de los interfaces de usuario orientada a aumentar la capacidad expresiva de los sistemas. El capítulo 4 concluye esta parte con el análisis del estado del arte presentado en los dos capítulos anteriores.

- ⇒ En la Parte II se presenta la contribución teórica de esta tesis. El capítulo 5 describe la visión de la interacción usuario-sistema que se plantea en esta tesis y los principios que han guiado el desarrollo de la misma en los entornos de toma de decisiones, marcando de esta manera las metas a conseguir en lo que respecta al nivel conceptual e instrumental de la conversación entre un sistema y sus usuarios, y el conjunto de requisitos del modelo perseguido. En el capítulo 6, se presenta una arquitectura que satisface dichos requisitos y soporta el tipo de interacción deseado, explicando con detalle cada uno de los componentes de dicha arquitectura.
- ⇒ En la Parte III se justifica la viabilidad del modelo propuesto mediante su aplicación al desarrollo de un sistema de ayuda a la decisión en la gestión de transporte público. El capítulo 7 presenta el diseño de esta aplicación y el capítulo 8 describe cómo se ha implementado con la herramienta KSM.
- ⇒ Por último, se incluye una sección de conclusiones con dos capítulos: el capítulo 9, donde se hace una evaluación de los logros alcanzados en la interacción usuario-sistema con la aplicación de la arquitectura propuesta en esta tesis, y el capítulo 10, que presenta las conclusiones obtenidas del trabajo desarrollado en la tesis, los objetivos alcanzados y las limitaciones observadas, para terminar con una relación de trabajos futuros.

Finalmente, la memoria se completa con información adicional relativa al interfaz del prototipo de gestión de transporte público, contenida en un anexo, y con la relación de las referencias bibliográficas aparecidas en el texto.



# Parte I: ANTECEDENTES





## **2. Interacción Usuario-Sistema Como Resultado De La Estructuración del Conocimiento**

---

Mejorar la calidad de la comunicación entre sistemas informáticos complejos, orientados a proporcionar ayuda en la gestión de un dominio problema, y sus usuarios, entendidos como operadores con diferentes niveles de capacitación y responsabilidad en ese dominio, requiere salvar la distancia que existe entre el modo en el que estos usuarios conciben el dominio problema y el modelo de comprensión de ese dominio que emplea el sistema en sus razonamientos. Una de las formas de acercarse a este objetivo es construyendo para los sistemas, modelos de conocimiento del dominio problema expresados en términos comprensibles para los usuarios, de manera que éstos puedan acceder, criticar o modificar el conocimiento de estos sistemas.

Las soluciones, que en este sentido aportó la primera generación de Sistemas Basados en el Conocimiento, resultan claramente insuficientes al carecer estos sistemas de las distintas clases de conocimiento necesario y de modelos de organización del mismo adecuados para poder dar a sus usuarios este tipo de facilidades. Fundamentalmente, los problemas surgían de la falta de:

- una formulación explícita de las estrategias de razonamiento del sistema en un lenguaje más abstracto que el que imponía un determinado formalismo de representación del conocimiento, tal como las reglas, los marcos o las restricciones, y
- una clara distinción entre los distintos tipos de conocimiento que se empleaban en ese razonamiento.



A mediados de los ochenta, la segunda generación de Sistemas Basados en el Conocimiento empezó a plantear soluciones a los problemas anteriores basadas en la búsqueda de un lenguaje abstracto de representación del conocimiento, alejado de su expresión computacional, que permitiese expresar los diferentes tipos de conocimiento de un sistema en términos cercanos a los usuarios. Por otra parte, el uso de este lenguaje impuso la necesidad de introducir estructura conceptual en la organización de todo este conocimiento del que ahora podían disponer los sistemas.

Este capítulo presenta las soluciones de modelización y organización del conocimiento más significativas, desarrolladas desde mediados de los ochenta, que han permitido en la actualidad elevar el nivel de cooperación entre usuarios y sistemas al mejorar su capacidad de comunicación.

## **2.1 La Modelización del Conocimiento**

A continuación se describen las propuestas más importantes de estructuración del conocimiento, en las que puede observarse como común denominador la descripción de los sistemas basados en el conocimiento a partir de sus objetivos y el conocimiento de que disponen para alcanzarlos, sin plantearse, en principio, cómo ha de ser su representación simbólica y procesable.

### **2.1.1 El Nivel de Conocimiento de Newell**

Newell, en su artículo *The Knowledge Level* [Newell, 82], dio un paso decisivo en la dirección anterior caracterizando a los sistemas basados en el conocimiento como agentes capaces de realizar tareas inteligentes. Los sistemas, así entendidos, podían describirse en función de sus objetivos y del conocimiento que utilizaban para conseguirlos sin comprometerse con ningún formalismo de representación. Así, se intentaba dar forma al conocimiento sobre cómo resolver problemas empleado por un experto a un nivel más abstracto del que ofrecían los modelos simbólicos, y en definitiva más realista por estar más próximo a las intuiciones de las personas, ya que los expertos en cualquier clase de problemas hablan de las tareas que realizan y lo que necesitan para poderlas llevar a cabo; y no en términos de determinadas combinaciones de formulaciones en reglas, marcos o restricciones.

Este nuevo nivel de especificación de sistemas informáticos, denominado *nivel de conocimiento*, se suma a los niveles tradicionales de especificación de aplicaciones informáticas, que pueden caracterizarse como nivel de dispositivo, de circuito, lógico, y simbólico; y se describe en sus mismos términos. Los niveles tradicionales describen un sistema a distinto grado de detalle, siendo el nivel de dispositivo el más específico y el nivel simbólico el más abstracto. Además, la especificación de un sistema a uno de estos niveles siempre puede transformarse en una descripción a niveles inferiores. Esto sigue siendo cierto para el nivel de conocimiento, el cual se sitúa en el nivel máximo de abstracción, por encima del nivel simbólico.

La consideración del nivel de conocimiento como un nivel más de especificación de sistemas se sustenta en su descripción en los mismos términos que los empleados para los niveles clásicos. Estos términos caracterizan a un nivel de especificación de un sistema a partir de: el propio *sistema*, un *medio* que es lo que se procesa, *componentes* que realizan inferencias básicas con el medio, *leyes de composición* que definen como ensamblar los componentes para construir el sistema, y *leyes de comportamiento* que determinan cómo el comportamiento del sistema se deriva del comportamiento de los componentes y de la estructura del sistema.

En el nivel de conocimiento, la descripción de una aplicación en los términos anteriores considera que (ver figura 2.1):

- el sistema es un *agente*,
- su medio es el *conocimiento*,
- sus componentes son un conjunto de *metas* u objetivos, un conjunto de *acciones* con las que se comunica con su entorno, y un *cuerpo de conocimiento* que incluye lo que sabe,
- tiene una única ley de comportamiento que es el *principio de racionalidad*, mediante el cual el agente siempre va a seleccionar aquellas acciones que le ayuden a conseguir sus objetivos, y
- no tiene leyes de composición.

Una de las diferencias más significativas de un sistema entendido como un agente al nivel de conocimiento, al compararlo con los demás niveles, es la absoluta falta de estructura en el caso del agente, no hay estructura en la organización de los componentes ni tampoco en los propios componentes. Esto



hace que el comportamiento del agente dependa sólo de lo que sabe, lo que quiere conseguir y los medios de que dispone para conseguirlo; mientras que en el caso de los demás niveles el comportamiento está determinado por la estructura del sistema.

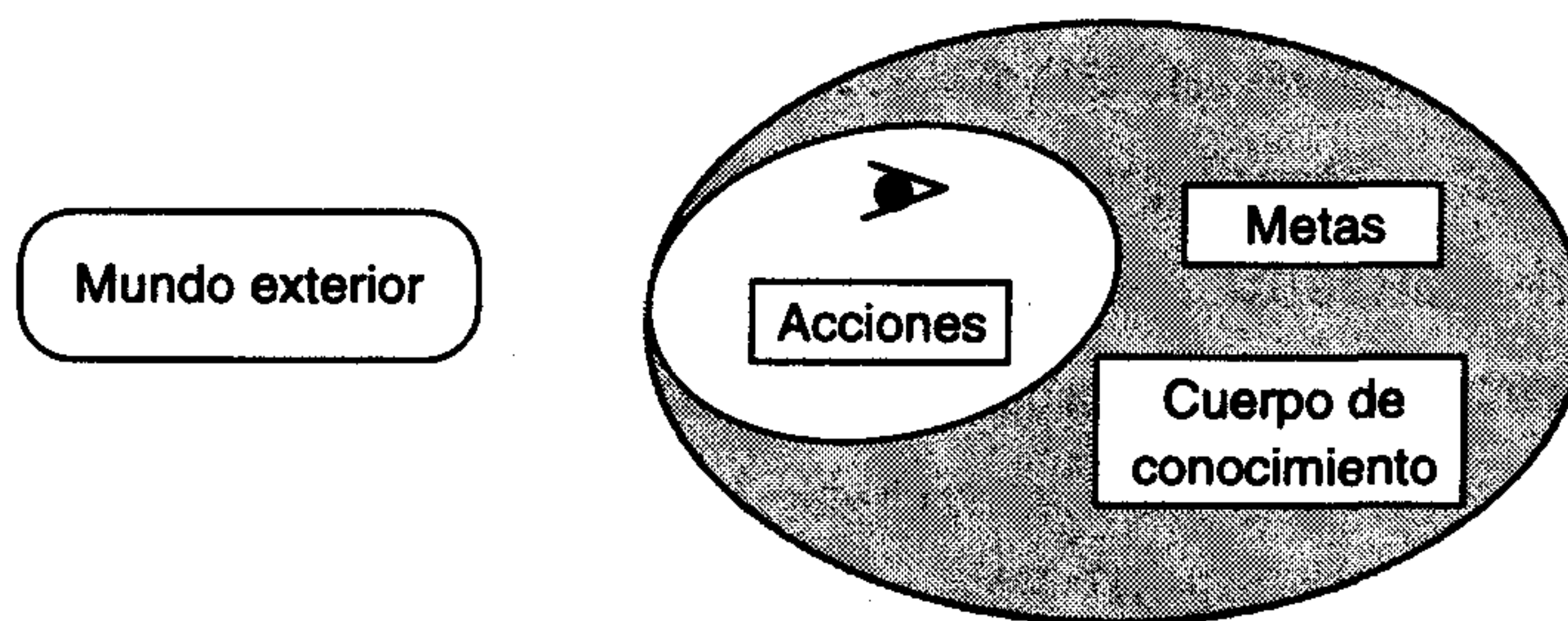


Figura 2.1: Elementos de un agente  
(adaptada de [Newell, 82])

Esta característica de la especificación de sistemas al nivel de conocimiento introdujo una modificación en la metodología de trabajo de los ingenieros del conocimiento por la que, antes de llevar a cabo el diseño informático y plantear el modelo computable de un sistema, había de realizarse una nueva etapa de análisis en la que se reflexionase sobre la forma de entender el problema.

### 2.1.2 El modelo de Tareas Genéricas

Del análisis de los sistemas inteligentes desde un nivel alejado de la implementación, Chandrasekaran [Chandrasekaran, 83, 86, 87] propuso un nuevo esquema de modelización del conocimiento soportado por lo que denominó *tareas genéricas*. Una tarea genérica identifica o caracteriza a un tipo de problema, un método o proceso de inferencia capaz de resolver el problema, y las distintas clases de conocimiento que necesita ese método.

Este concepto surge de la observación de que un dominio problema puede descomponerse de forma natural en subestructuras, cada una de las cuales especializada en la resolución de una clase de problemas. A su vez, estas subestructuras pueden descomponerse en otras más simples especializadas en distintos conceptos de la misma clase de problemas. Así, un sistema complejo podría concebirse como una organización modular y jerarquizada de tareas



genéricas, cada una de las cuales estaría especializada en la solución de una parte del problema resuelto por el sistema.

La primera tarea genérica que identificaron Chandrasekaran y su grupo fue la tarea de *clasificación jerárquica* [Chandrasekaran, Mittal, 83] que consiste en identificar o clasificar un objeto basándose en un conjunto de aspectos. El conocimiento que utiliza esta tarea está representado como una jerarquía de clases y subclases sobre la que se realiza una búsqueda empleando la estrategia de *establecer & refinar*. El método de *establecer & refinar* comienza por intentar establecer, o confirmar con un grado de similaridad suficiente, la clase de más alto nivel. Si ésta se confirma entonces se pasa a la etapa de refino en la que sus subclases pasan a ser hipótesis a considerar en el siguiente paso de inferencia. Si una clase no puede establecerse entonces tampoco se refina.

La descripción a este nivel de generalidad de la tarea de clasificación hace posible su aplicación en la resolución de problemas, como por ejemplo el diagnóstico, en diferentes dominios, como enfermedades o fallos mecánicos. En el caso del diagnóstico médico, la tarea recibiría como entrada un conjunto de síntomas y tendría que identificar la enfermedad que provoca dichos síntomas. El conocimiento en este caso sería una jerarquía de clases de enfermedades, las más generales en los niveles superiores de la jerarquía y las más específicas en los inferiores. El control del sistema está descrito explícitamente al nivel de tarea, esto es, las clases se evalúan y después si es necesario se refinan; y no al nivel de implementación o simbólico.

La tarea genérica plantea un modelo de organización radicalmente distinto al empleado tradicionalmente, en el que el conocimiento de un sistema se encontraba aislado en una base de conocimiento a la que podían acceder distintos métodos de inferencia. Por contra, con las tareas genéricas, conocimiento e inferencia se agrupan en función del tipo de problema que resuelven. De esta manera, aunque parece perderse la generalidad del enfoque tradicional, se ofrece una visión más clara y realista de la organización del conocimiento de resolución de problemas en un dominio complejo.

El concepto de tarea genérica se mostró insuficiente para reflejar el comportamiento inteligente que cabe esperar de una aplicación compleja basada en el conocimiento. La tarea genérica liga el objetivo o problema a resolver con un modo particular de alcanzarlo, mientras que el

comportamiento inteligente se caracteriza por la posibilidad de explorar alternativas de solución. Es por ello que la tarea genérica evolucionó para acercarse más a esta forma de entender la inteligencia ([Chandrasekaran et al., 92]), que se halla más próxima a las ideas de Newell, separando la tarea o actividad a realizar del método que se emplea para llevarla a cabo, que puede ser más de uno, y entendiendo que la aplicación del método puede implicar la realización de varias subtareas. Por ejemplo, en la figura 2.2 se muestran tres métodos alternativos para llevar a cabo la tarea de diagnóstico: un método bayesiano que aplica técnicas estadísticas para deducir el diagnóstico, un método abductivo que obtiene el diagnóstico como resultado de una agregación de hipótesis, y el método *cubrir & diferenciar* que se describe en el apartado 2.1.3.

Con este nuevo modo de entender la organización de una tarea genérica, la estructura del conocimiento de un sistema inteligente que pueda realizar la tarea compleja queda caracterizada como una jerarquía de tareas y métodos que opcionalmente pueden llevarlas a cabo, como se muestra en la figura 2.2.

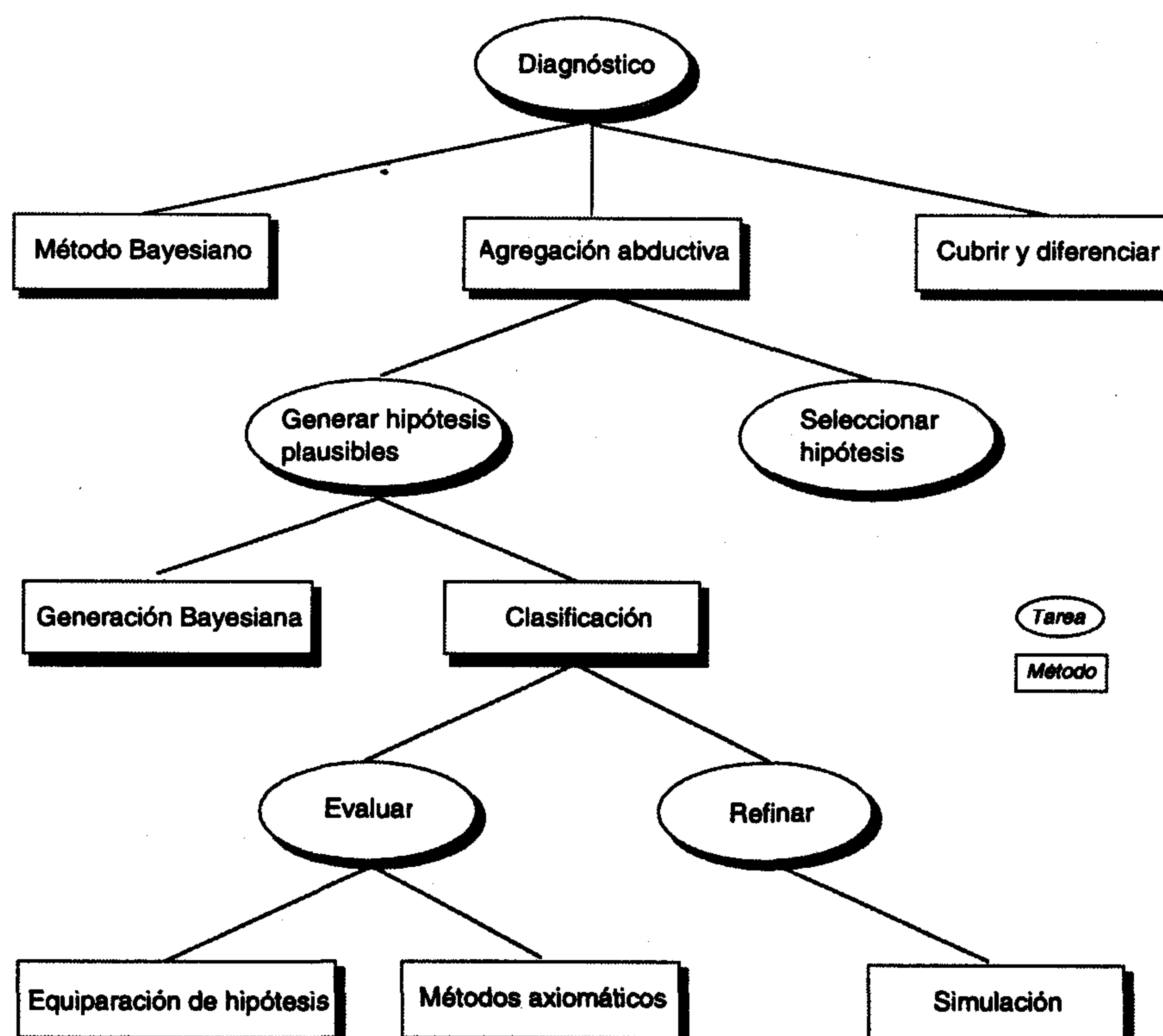


Figura 2.2: Estructura parcial de tareas y métodos para el problema de diagnóstico (adaptada de [Chandrasekaran et al., 92])



Las ventajas fundamentales de este nuevo enfoque de organización del conocimiento pueden resumirse como sigue:

- a nivel conceptual, la solución de un problema o una tarea no se aborda con un único método de resolución sino que se puede disponer de varios métodos alternativos, caracterizados por el conocimiento que necesitan, que hacen posible resolver el problema adaptándose a las peculiaridades de un dominio concreto.
- la visión estructurada de las tareas y los métodos junto con el conocimiento que necesitan, facilita el proceso de adquisición del conocimiento.
- la modularidad de la estructura de tareas, donde los distintos métodos quedan caracterizados explícitamente, hace posible la integración de resultados obtenidos por métodos basados en conocimiento cualitativo, cuantitativo, métodos algorítmicos y heurísticos, etc.

### 2.1.3 Los métodos “Role Limiting”

Las técnicas de representación del conocimiento e inferencia empleadas hasta principios de los ochenta eran aplicables en la realización de un amplio rango de tareas debido a su gran generalidad, pero esta misma generalidad hacía que se perdiesen las peculiaridades de los distintos tipos de problemas que se resolvían. A la vista de este hecho, y con el objetivo de desarrollar herramientas de adquisición del conocimiento, McDermott y su equipo [McDermott, 88] se plantearon la necesidad de introducir en la construcción de sistemas basados en el conocimiento métodos de resolución de problemas especializados en tipos de tareas a los que denominó métodos *role limiting*.

En los métodos *role limiting* se representa el hecho de que casi todo el conocimiento que un método utiliza para identificar e implementar las acciones que realiza, así como para seleccionar entre acciones candidatas, es conocimiento específico de la tarea o problema que resuelve. Así, al identificar el rol que juega cada pieza de conocimiento en el proceso de razonamiento, el método puede facilitar el proceso de adquisición de conocimiento ya que se produce una clasificación del conocimiento según el papel que juega en los

diversos pasos de inferencia y, al mismo tiempo, el rol identificado orienta sobre la forma de representar ese conocimiento.

Este concepto de método de resolución de problemas fue empleado en el desarrollo de diversas herramientas de adquisición del conocimiento entre las que se encuentran MOLE [Eshelman et al., 87] y SALT [Marcus, McDermott, 89]. MOLE es una herramienta para construir sistemas de diagnóstico basada en el método *role-limiting* denominado *cubrir & diferenciar*. Para que MOLE resulte adecuado a la hora de resolver un problema de diagnóstico, es necesario que en el dominio de aplicación se disponga de conocimiento sobre síntomas o manifestaciones de problemas, causas que puedan explicar la presencia de estos síntomas y criterios que ayuden a discriminar entre explicaciones candidatas. Sobre este conocimiento, el control del método establece que primero se determinen los eventos que cubren o explican los síntomas observados para después descartar o diferenciar aquellos que no resultan verosímiles.

MÉTODO: PROPONER & REVISAR
<ol style="list-style-type: none"><li>1. Proponer un paso de diseño e identificar restricciones sobre ese diseño propuesto.</li><li>2. Identificar restricciones no satisfechas. Si no hay ninguna, volver al punto 1.</li><li>3. Plantear soluciones posibles a una restricción no satisfecha.</li><li>4. Elegir la solución menos costosa de entre las que no han sido ya probadas.</li><li>5. Modificar el diseño de acuerdo con la solución adoptada e identificar restricciones sobre la modificación que se acaba de introducir.</li><li>6. Identificar restricciones no satisfechas debido a la modificación anterior. Si hay alguna volver al punto 4.</li><li>7. Eliminar las relaciones incompatibles con la revisión del diseño.</li><li>8. Si el diseño está incompleto, volver al punto 1.</li></ol>

Figura 2.3: Método *role-limiting* proponer & revisar

Por su parte, SALT es una herramienta de construcción de sistemas de diseño que se basa en el método *role-limiting* llamado *proponer & revisar* (ver figura 2.3). La aplicabilidad del método *proponer & revisar* depende de la disponibilidad de procedimientos que proporcionen valores iniciales para las piezas del objeto de diseño, procedimientos que definan restricciones sobre los valores que pueden incorporarse en el diseño de esas piezas y soluciones a la violación de estas restricciones. El conocimiento de control impone que



primero se generen propuestas de diseño que posteriormente sean revisadas de acuerdo con un conjunto de restricciones de diseño prefijadas.

Estos dos ejemplos ilustran la naturaleza del conocimiento de control de un método *role-limiting*, que es útil para decidir qué paso de inferencia hay que realizar en cada momento dentro del contexto de un tipo concreto de tarea.

Con la definición de los métodos *role-limiting* se puso de manifiesto la posibilidad de mejorar las herramientas de adquisición de conocimiento existentes ya que al elegir uno de estos métodos se sabe a priori:

- qué conocimiento específico va a necesitarse para resolver una determinada tarea, lo que representa una guía a la hora de obtener ese conocimiento de un experto.
- cómo va a utilizarse para encontrar la solución, lo que facilita la elección del modo más adecuado de representar ese conocimiento.

Por otra parte, la aplicación de este tipo de métodos permite dar un paso más en el acercamiento del usuario al sistema al mejorar el nivel de explicación que puede ofrecer un sistema sobre sus conclusiones, ya que puede justificar el uso del conocimiento en cada etapa al conocer cuál es el papel que éste juega en el razonamiento.

### 2.1.4 Los niveles de conocimiento de KADS

KADS (*Knowledge Acquisition and Design System*) [Wielinga et al., 92] propone un marco conceptual para la modelización de conocimiento experto basándose en dos premisas fundamentales:

- (i) que es posible distinguir entre varios tipos de conocimiento atendiendo a los diferentes roles que ese conocimiento puede jugar en los procesos de razonamiento y,
- (ii) que estos tipos de conocimiento pueden organizarse en varios niveles entre los que hay una limitada interacción.



Atendiendo a estas premisas, KADS propone una clasificación del conocimiento a cuatro niveles que amplía la clásica diferenciación entre conocimiento del dominio y conocimiento de control:

- el *nivel de dominio*, que define una teoría declarativa del dominio de la aplicación y en el que están incluidos los conceptos, propiedades de los mismos y relaciones entre ellos, así como conceptos agregados o estructuras;
- el *nivel de inferencia*, donde se describen las inferencias que pueden realizarse con el conocimiento del nivel de dominio. En este nivel se distinguen tres tipos de conocimiento:
  - ◊ *fuentes de conocimiento* o pasos elementales de inferencia,
  - ◊ *metaclases* o clases de elementos sobre los que operan las fuentes de conocimiento, y
  - ◊ *visiones del dominio* que especifican qué parte del conocimiento del dominio puede ser usada por una fuente de conocimiento.
- el *nivel de tareas*, donde se especifican las formas en que han de combinarse las fuentes de conocimiento para definir estructuras de inferencia que permitan alcanzar los objetivos perseguidos. En este nivel también se distinguen tres tipos de conocimiento:
  - ◊ *tareas* o acciones compuestas de resolución de problemas,
  - ◊ *términos de control* o vocabulario utilizado para dar nombre a conjuntos de elementos de metaclases, y
  - ◊ *estructura de una tarea* o descomposición de una tarea en subtareas y especificación de las dependencias de control entre esas subtareas.
- el *nivel estratégico*, donde se especifican los objetivos a alcanzar para resolver un problema particular mediante las tareas, es decir, donde se establecen dinámicamente los planes de ejecución de las tareas. Este nivel, aunque aporta flexibilidad al modelo no ha sido desarrollado al mismo nivel de detalle que los anteriores por lo que su aplicación práctica, en la mayoría de los casos, se ha limitado al uso de estructuras de tarea predefinidas y no modificables en tiempo de ejecución.

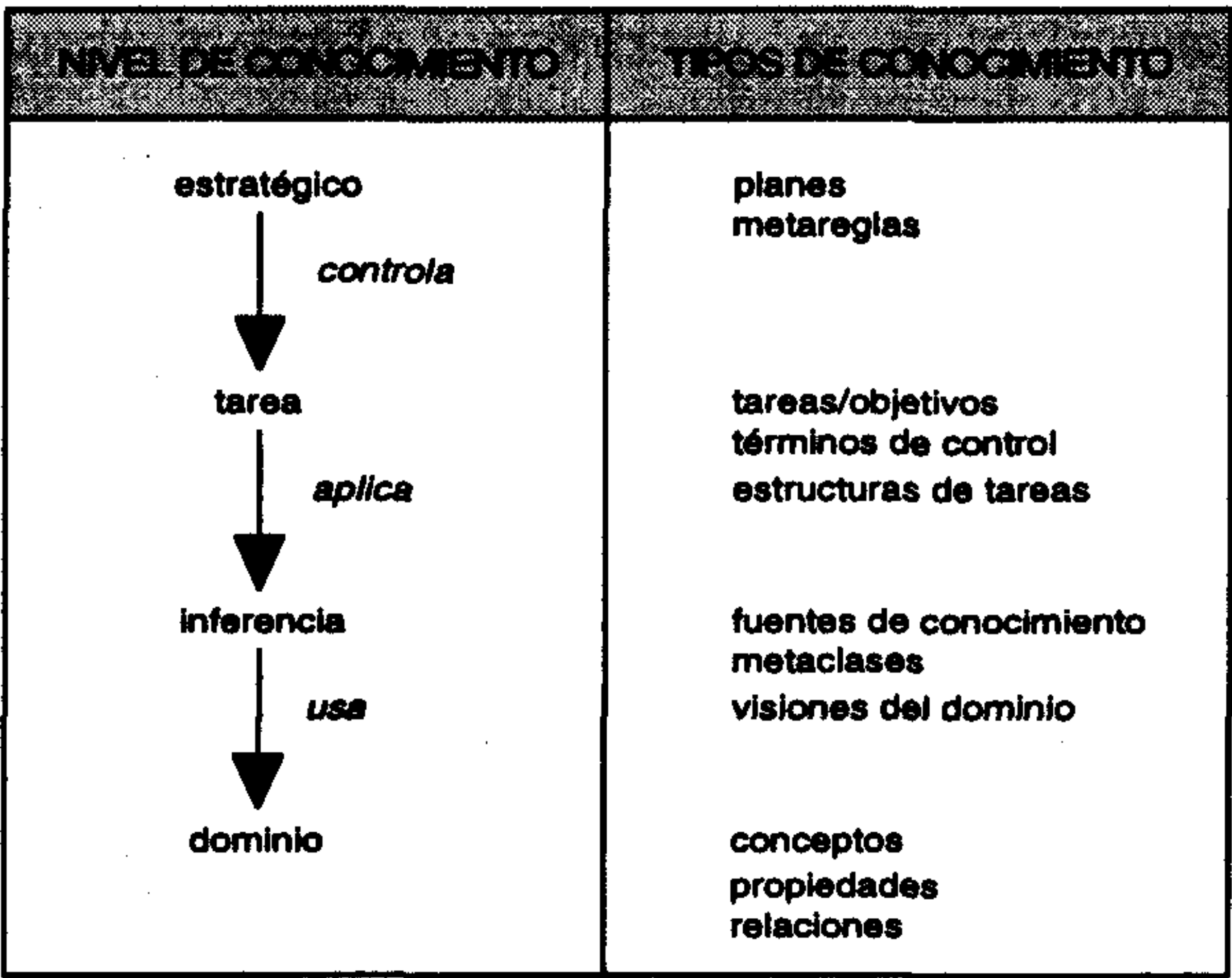


Figura 2.4: Modelo a cuatro niveles de KADS

Por otra parte, KADS permite disponer de una librería de estructuras de razonamiento generales para clases de problemas, llamadas modelos de interpretación, en las que se describen los niveles de inferencia y tarea pero no el de dominio, haciendo posible así su reutilización en diferentes dominios.

En resumen, KADS es una forma de dar estructura al nivel de conocimiento de Newell basada en la identificación de cuatro categorías de conocimiento.

2.1.5 El modelo de Componentes de la Experiencia

Luc Steels [Steels, 90, 92] planteó un modelo de diseño de sistemas inteligentes al nivel de conocimiento formulado desde tres perspectivas correspondientes a las clases de componentes de la experiencia identificables en un modelo de conocimiento experto: modelos, tareas y métodos.

En la *perspectiva de modelos*, cada modelo representa una abstracción de una parcela de la realidad en la que se identifican objetos, sus propiedades y atributos, relaciones entre propiedades de objetos, etc., dando forma así a una estructura de la información disponible en un dominio. Los modelos pueden ser de dos tipos: de dominio o de caso. Los primeros representan información estática, no modificable con los procesos de razonamiento, que es válida para diferentes aplicaciones que se desarrollen en ese mismo dominio. Los segundos incluyen datos válidos solamente para una aplicación concreta que se modifican con los procesos de inferencia.



En la *perspectiva de tareas*, cada tarea representa un objetivo a alcanzar actuando sobre uno o más modelos. Las tareas se organizan en una estructura jerárquica obtenida de la descomposición de tareas de alto nivel en otras subtareas más sencillas hasta alcanzar tareas elementales, de forma análoga a como se hace con las tareas genéricas. Las tareas, como los modelos, pueden ser de dos tipos:

- (i) *tareas de adquisición del conocimiento* empleadas únicamente en la fase de adquisición del conocimiento del sistema con el objetivo de construir modelos de dominio; y
- (ii) *tareas de aplicación* empleadas durante la ejecución del sistema y responsables de generar modelos de caso.

En la *perspectiva de métodos*, cada método especifica la forma en que se lleva a cabo una tarea mediante una serie de actividades y un flujo de control de las mismas. Según el tipo de tarea asociada a un método, se tendrán métodos de tareas descomponibles y métodos que realizan tareas elementales.

En resumen, Steels ofrece una metodología de construcción de sistemas inteligentes que partiendo de la tarea de más alto nivel asociada al problema que ha de resolver el sistema realiza un refinamiento jerárquico de la misma, hasta alcanzar las tareas elementales, en el que en cada paso de refinamiento la tarea analizada se caracteriza desde las perspectivas de modelo y método.

### 2.1.6 El modelo de Unidades Cognitivas

[Cuenca, 93a], [Molina, 93], [Cuenca, Molina, 94, 96, 97] plantearon la unidad cognitiva como pieza básica de estructuración de conocimiento que tiene como objetivo la modelización de un área de conocimiento de un experto en un dominio determinado. Este modelo de organización incluye dos partes claramente diferenciadas (ver figura 2.5):

- la descripción de su funcionalidad, esto es, qué tareas puede llevar a cabo la unidad, y
- la descripción de los componentes o áreas de conocimiento de que dispone la unidad para realizar esas tareas.

Desde este punto de vista, una unidad cognitiva es una representación de un agente tipo Newell en la que lo que el agente sabe se define con áreas de conocimiento y lo que el agente es capaz de hacer se define con tareas. Sin embargo, a diferencia del planteamiento de Newell, las unidades cognitivas que describen a los agentes pueden descomponerse en otras unidades cognitivas más sencillas, ofreciendo así una vía para estructurar el diseño de los sistemas.

Los componentes de conocimiento, estructurados a su vez como unidades cognitivas, ofrecen el soporte necesario para llevar a cabo las subtarefas asociadas a los métodos que desarrollan la funcionalidad de la unidad cognitiva. Estos componentes, a su vez, pueden definir áreas de conocimiento y por tanto representarse con unidades cognitivas, desarrollando una estructura jerárquica en cuyas hojas se encuentran las denominadas unidades cognitivas primarias, es decir, áreas que ya no se descomponen y que pueden ser modelizadas directamente con técnicas básicas de representación del conocimiento. Un ejemplo de área de conocimiento sería aquella que incluyese conocimiento sobre enfermedades infecciosas, la cual podría dividirse en otras dos más específicas: área con conocimiento sobre detección y diagnóstico de enfermedades infecciosas, y área con conocimiento para prescribir tratamientos a determinadas enfermedades infecciosas.

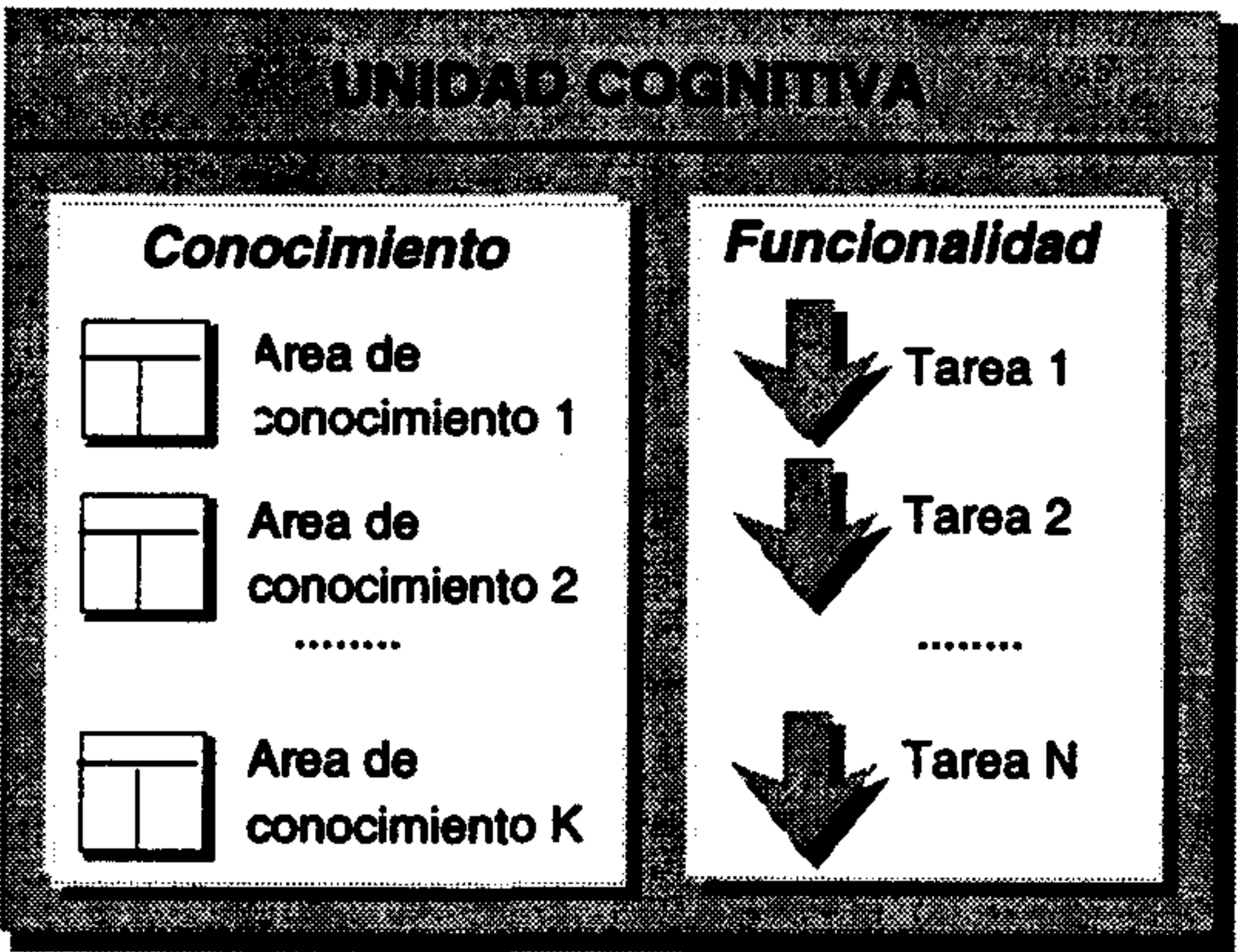


Figura 2.5: Estructura de una unidad cognitiva

Cada tarea de una unidad cognitiva define una acción de resolución de problemas soportada por un área de conocimiento. La realización de una tarea la lleva a cabo un método de resolución de problemas que determina una estrategia de razonamiento a partir de subtarefas más sencillas. De esta manera,



para cada tarea llega a definirse un árbol de tareas-subtareas en cuyas hojas se encuentran las tareas asociadas a las unidades cognitivas primarias. Así, cada unidad cognitiva tendrá asociados tantos árboles de este tipo como tareas pueda realizar dentro de su área de conocimiento. Por ejemplo, las tareas diagnosticar enfermedades y diseñar tratamiento son inherentes al área de conocimiento sobre enfermedades infecciosas, las cuales podrían resolverse con métodos como los que se han descrito en secciones anteriores, esto es, *cubrir & diferenciar* para la de diagnóstico y *proponer & revisar* para la de diseño.

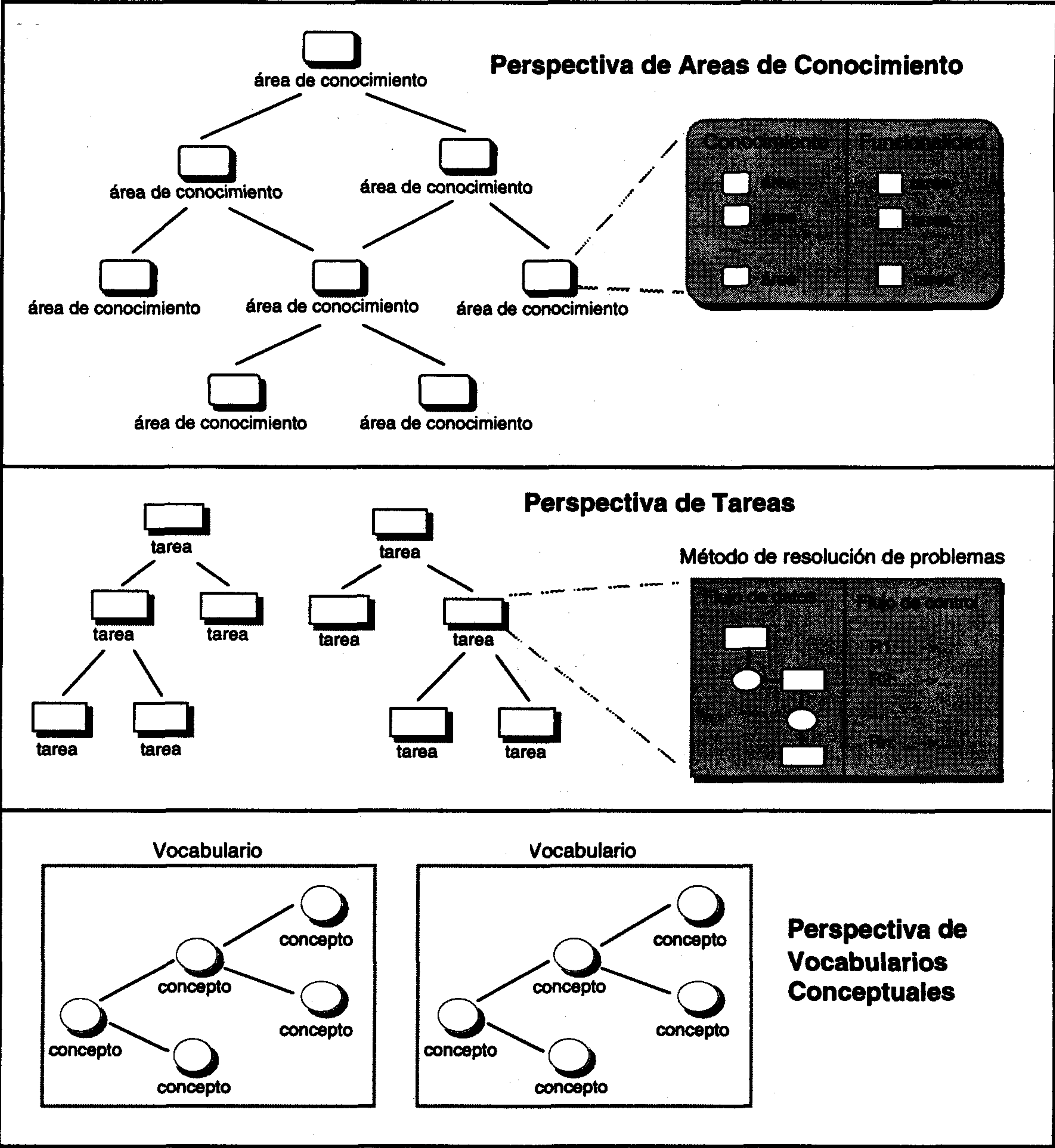


Figura 2.6: Perspectivas de un modelo de conocimiento basado en unidades cognitivas



El tipo de estructuración del conocimiento planteado por las unidades cognitivas hace necesario introducir mecanismos que aseguren el entendimiento entre módulos distintos que manejen una terminología común. Esto se lleva a cabo mediante lo que se denominan vocabularios conceptuales, los cuales definen conjuntos de conceptos comunes a diferentes unidades cognitivas primarias. Un vocabulario conceptual se utiliza para definir el conocimiento local a una unidad primaria, pero no es local a esa unidad ya que puede estar compartido por otras.

Por tanto, el paradigma de modelización con unidades cognitivas incorpora tres perspectivas inter-relacionadas de un modelo de conocimiento que pueden hacerse explícitas a partir de la definición de cada unidad cognitiva (ver figura 2.6):

- la *perspectiva de áreas de conocimiento*, donde se ofrece una visión modular y declarativa de la organización del conocimiento;
- la *perspectiva de tareas*, donde se representan los diferentes tipos de razonamiento que se realizan para resolver problemas;
- la *perspectiva de vocabularios conceptuales*, donde se define la terminología común a distintas áreas de conocimiento.

En resumen, el modelo de estructuración del conocimiento basado en unidades cognitivas ofrece una representación intuitiva y de alto nivel del modo de entender una clase de problemas que facilita su comprensión y permite participar en el diseño a usuarios no programadores capaces de organizar de esta manera su conocimiento.

## 2.2 Modelos Estáticos de Conocimiento

Las propuestas de modelización del conocimiento presentadas anteriormente ofrecen soluciones al problema de caracterizar o expresar el conocimiento de un dominio de forma alejada de los aspectos de implementación. Ahora bien, la solución efectiva de un problema requiere que las estructuras de conocimiento derivadas de los modelos anteriores se empleen de acuerdo con unas

determinadas directrices de control que definen la forma en que finalmente ha de alcanzarse la solución a los problemas a resolver con el modelo que se formula.

Aunque los modelos de conocimiento presentados anteriormente son conceptualmente muy flexibles en lo que se refiere a los modos de llevar a cabo las diferentes tareas de una aplicación, es decir, consideran que un mismo problema o tarea puede resolverse de formas distintas, las realizaciones prácticas y las herramientas desarrolladas con estos modelos están basadas en relaciones unívocas entre tareas y métodos, haciendo que la elección del método adecuado para una tarea se tome durante la etapa de diseño. En esta sección se presentan algunos de estos resultados.

### 2.2.1 KREST

El modelo de componentes de la experiencia de Steels se plasmó en una herramienta de construcción de modelos a nivel de conocimiento llamada KREST (*Knowledge Reusable Tool*) [Macintyre, 93] orientada a facilitar a usuarios no programadores la construcción y mantenimiento de sistemas inteligentes. Para ello ofrece la posibilidad de describir a alto nivel las aplicaciones, sin considerar los aspectos de implementación, mediante un entorno gráfico amigable y, una vez descritas, traducirlas automáticamente a un sistema ejecutable mediante la utilización de métodos solución previamente definidos y almacenados en una biblioteca.

El contenido de esta biblioteca puede ser ampliado por el desarrollador de una aplicación programando nuevos métodos, que necesite y no estén disponibles en la misma, y que posteriormente pueden ser utilizados para construir otros sistemas. Por otra parte, el usuario final de una aplicación puede entender el modo de razonar del sistema ya que tanto el conocimiento del mismo como su estructura de razonamiento son accesibles y están expresados en términos de alto nivel, más comprensibles para usuarios no programadores, lo cual aumenta significativamente la usabilidad del sistema.

El entorno gráfico a partir del que se construyen los sistemas maneja tres tipos de herramientas descriptivas (ver figura 2.7):

- el *árbol de tareas* permite presentar la estructura de la aplicación mediante la instrumentación del mecanismo de refinamiento jerárquico de tareas presente en el modelo de componentes de la experiencia;
- el *diagrama de dependencia de modelos* muestra la estructura funcional de la aplicación mediante una representación semejante a un diagrama de flujo de datos en el que los modelos actúan como abstracciones de estructuras de información y los métodos reciben y generan la información conforme con los modelos;
- el *diagrama de control* describe el comportamiento secuencial de los métodos mediante un conjunto de estados, cada uno de los cuales es un método, entre los que se transita utilizando información de control, que expresa el resultado de la ejecución del método origen de la transición. Así, desde el punto de vista del control, los métodos generan estados que determinan qué métodos habrá que ejecutar a continuación.

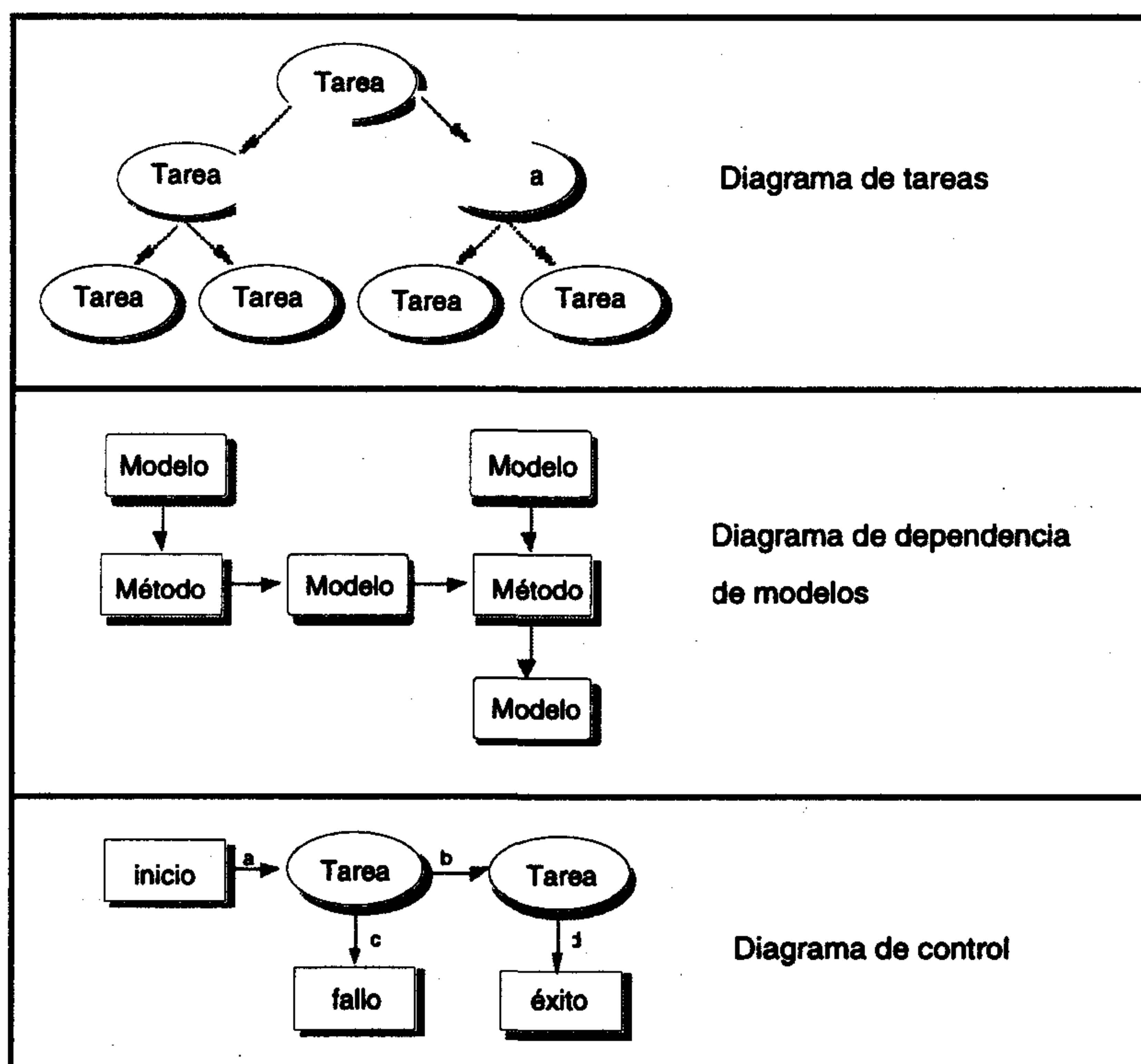


Figura 2.7: Herramientas de modelización de KREST



El desarrollo de una aplicación parte de la definición de la tarea de más alto nivel, que representa a toda la aplicación, a la que se va descomponiendo de arriba a abajo en forma de árbol hasta alcanzar los nodos hoja que serán aquellos a los que se asociarán métodos solución de la biblioteca. En cada nodo de este árbol se lleva a cabo la descripción del correspondiente diagrama de dependencia de modelos y diagrama de control. Finalmente, se deciden los formalismos de representación más adecuados para los modelos y los métodos entre diferentes opciones proporcionadas por KREST y se genera la versión ejecutable de la aplicación.

## 2.2.2 PROTÉGÉ-II

PROTÉGÉ [Musen, 89] [Tu et al., 92] [Puerta et al., 93], es un entorno de desarrollo de modelos de conocimiento sobre problemas de planificación procesables mediante el método de refinamiento progresivo de planes que está en la línea de los trabajos dedicados a facilitar los procesos de adquisición del conocimiento mediante la identificación de modelos abstractos de resolución de problemas como los que dieron lugar a los sistemas ROGET [Bennet, 85], SALT [Marcus, McDermott, 89] y MOLE [Eshelman et al., 87].

De la experiencia con el sistema PROTÉGÉ, el mismo grupo desarrolló PROTÉGÉ-II [Puerta et al., 93], [Musen et al., 95], [Eriksson et al., 95], [Tu et al., 95] como una herramienta de adquisición del conocimiento no dependiente de un modelo de razonamiento específico sino orientada a la formulación de cualquier tipo de método como eje del razonamiento de una clase de aplicaciones. Una vez planteado un problema, el ingeniero de conocimiento utiliza PROTÉGÉ-II para construir un método genérico capaz de resolver dicho problema a partir de la composición de piezas reutilizables alojadas en una biblioteca. El sistema final se obtendrá de la instanciación de dicho método genérico con el conocimiento de un dominio concreto. Por tanto, PROTÉGÉ-II se utiliza para construir herramientas de adquisición del conocimiento asociadas a métodos de resolución de problemas fijos que pueden instanciarse en distintos dominios.

Los elementos fundamentales de la arquitectura PROTÉGÉ-II son la *tarea*, el *método* y el *mecanismo* (ver figura 2.8). Una tarea es una abstracción de una actividad del mundo real en la que se especifican los objetivos que quieren obtenerse a partir de unas determinadas entradas pero no se compromete el

modo de alcanzar dichos objetivos. La forma de alcanzar los objetivos de las tareas la definen los métodos. Un método es un procedimiento descrito como una secuencia de pasos de inferencia que pueden a su vez ser tareas y por tanto resolverse con otros métodos, lo que describe un proceso de descomposición que culmina en tareas capaces de resolverse directamente por un método no descomponible, esto es, un mecanismo. Los mecanismos constituyen las piezas de software reutilizable cuya composición permite construir los métodos de resolución de problemas.

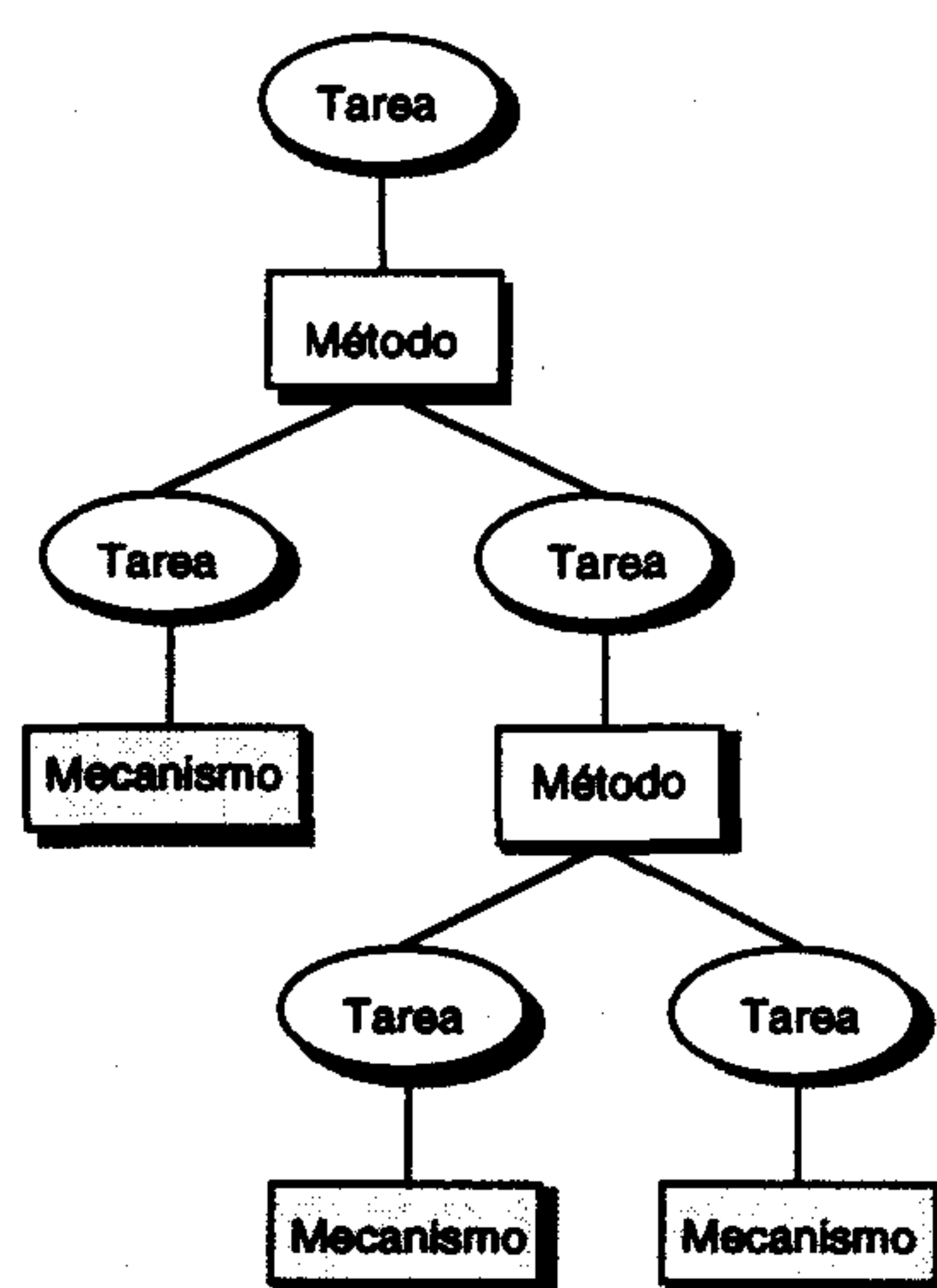


Figura 2.8: Estructura de tareas, métodos y mecanismos

Los elementos que definen un mecanismo son: sus entradas y salidas, el modelo de datos con el que se representan esas entradas y salidas, el modelo de control que describe cómo obtener las salidas a partir de las entradas y una especificación del interfaz con el que el mecanismo se comunica con el usuario. Por su parte, los métodos, además de los elementos de especificación definidos para los mecanismos, incluyen un modelo de descomposición de tareas, es decir, enumeran las tareas que han de resolverse durante la ejecución del método.

El proceso de construcción de un sistema con PROTÉGÉ-II tiene dos pasos fundamentales: configuración de métodos y ensamblaje de métodos. En la etapa de configuración se parte de un método y se eligen mecanismos para resolver las subtareas en las que se descompone el método haciendo que el modelo de datos de los mecanismos asuma el modelo de datos del método y construyendo el interfaz del método con la composición de los interfaces de los mecanismos. En la etapa de ensamblaje se crean nuevos métodos a partir de las



subtareas que han de componerlo, definiendo con ellas el modelo de datos, las entradas y salidas del método y el flujo de control para la ejecución de las subtareas.

Este esquema, por tanto, ofrece un interfaz que permite configurar y mantener sistemas basados en el conocimiento por usuarios expertos en el dominio problema. Constituye por tanto un salto cualitativo en la interacción hombre-máquina ya que la persona accede al modelo conceptual de una aplicación mediante un interfaz diseñado a la medida de las necesidades de diseño de modelos de conocimiento.

### 2.2.3 KSM

KSM (*Knowledge Structure Manager*) [Molina, 93], [Molina, Cuenca, 95], [Cuenca, Molina, 94, 96, 97] es una herramienta de construcción e instrumentación de modelos de conocimiento basados en unidades cognitivas que introduce una metodología de diseño de aplicaciones informáticas en la que se distinguen tres etapas:

- *Diseño del modelo conceptual.* En esta etapa se construye un modelo genérico a nivel de conocimiento que muestra la morfología del conocimiento y el rol que desempeña durante el razonamiento orientado a la resolución del problema. Es por tanto un modelo independiente del dominio e independiente del tipo de representación simbólica que se utilice finalmente para expresar el conocimiento.

Tras analizar el problema a resolver se define una estructura jerárquica con el conjunto de tareas a realizar y los métodos que pueden llevarlas a cabo. En el nivel inferior se encuentran aquellas tareas tratables o bien con métodos básicos genéricos de tipo clasificativo o constructivo, o bien con métodos ad hoc propios de la especialidad en estudio. De la observación de los cuerpos de conocimiento del dominio que soportan los distintos métodos se definen unidades cognitivas a distintos niveles agrupando aquellos métodos que hacen uso de idénticas estructuras de dominio. La jerarquía de unidades cognitivas que se obtiene es, por tanto, una síntesis de la jerarquía de tareas-métodos y sus estructuras de conocimiento de dominio asociadas cuyos nodos básicos son unidades cognitivas primarias.



- *Diseño del modelo simbólico.* En esta etapa se deciden tanto los formalismos de representación como los procedimientos de inferencia que se utilizarán con las unidades cognitivas identificadas previamente. El primer paso es identificar los vocabularios conceptuales locales a las distintas unidades primarias y extraer de ellos los conceptos comunes con otras unidades para definir vocabularios conceptuales globales. La definición de estos vocabularios se hace utilizando el lenguaje CONCEL [Molina, 93] [Sierra, Molina, 98]. A continuación, utilizando estos vocabularios se formulan las bases de conocimiento declarativo. Finalmente, recorriendo la jerarquía de abajo a arriba, se definen los métodos compuestos de niveles superiores a partir de los de niveles inferiores mediante un lenguaje de reglas de producción llamado LINK [Sierra, 95] [Molina et al., 98].

Esta etapa se ve facilitada por la disponibilidad de una biblioteca de piezas de software reutilizables, denominadas primitivas de representación, que implementan técnicas de resolución de problemas básicas (encadenamiento hacia delante de reglas, equiparación de marcos, propagación de restricciones, etc.) y que se asocian a las unidades cognitivas primarias. Como resultado de esta etapa se obtiene un modelo de resolución de problemas para una clase de aplicaciones con el que pueden generarse, mediante su instanciación en distintos dominios, distintos sistemas.

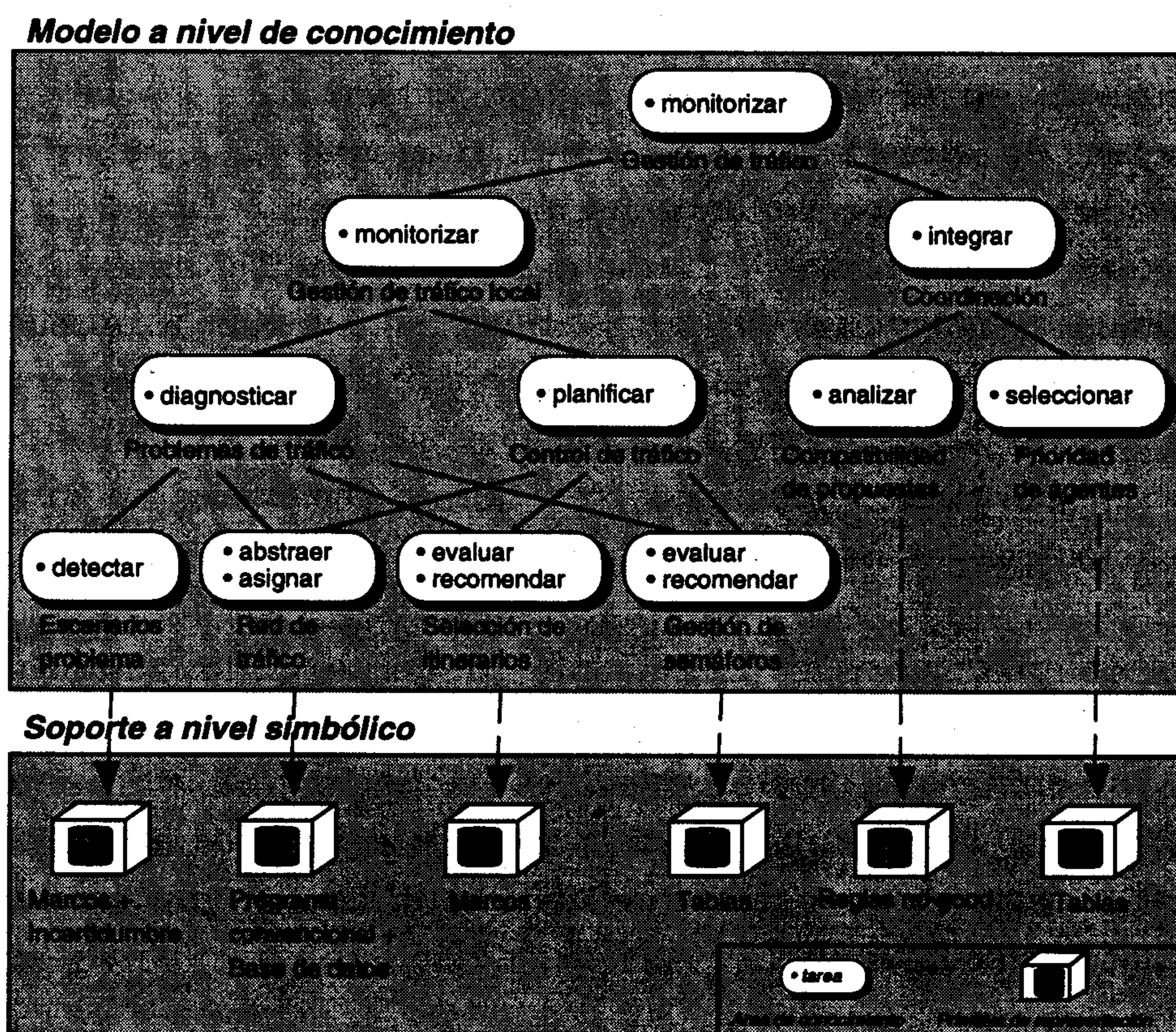
Por ejemplo, un modelo genérico de control de tráfico privado como el que se observa en la figura 2.9 concibe la monitorización del tráfico desde la integración de propuestas locales de control para configurar propuestas globales para toda la red. La gestión de áreas locales de tráfico emplea conocimiento de identificación de problemas y conocimiento de control.

La identificación de problemas requiere conocer la topología del área, el tipo de problemas que pueden presentarse, los criterios de selección de itinerarios de los conductores y la influencia de los planes de señalización con semáforos. Estas áreas de conocimiento se modelizan con unidades cognitivas primarias, cada una de ellas asociada a una primitiva de representación que determina el tipo de representación en el



que expresar el conocimiento del área y el tipo de inferencia que se va a aplicar.

La selección de planes de control local utiliza la información sobre la topología del área, los criterios de selección de itinerarios y el efecto de los planes de señalización con semáforos para, a partir de los problemas detectados, influir en la elección de itinerarios de los conductores y hacer que se alejen de la zona problemática. Finalmente, la integración de las diferentes propuestas locales se realiza tras eliminar posibles incompatibilidades y atendiendo a criterios de prioridad entre áreas problema.



*Figura 2.9: Asociación de primitivas de representación a un modelo de conocimiento de gestión de tráfico privado*

- **Instanciación del modelo genérico.** Durante esta fase el desarrollador, o en su caso el usuario que realiza el mantenimiento o extensión del conocimiento de la aplicación, introduce en las bases de conocimiento del modelo información concreta del dominio de acuerdo con la estructura y



representación establecida en la etapa anterior. Una vez hecho esto, el modelo está listo para ser ejecutado.

La ventaja de este enfoque de diseño de aplicaciones es doble: (i) por una parte, el modelo de estructuración del conocimiento con unidades cognitivas facilita el proceso de adquisición del conocimiento al orientarlo hacia el 'relleno' de bases específicas de conocimiento, y por otra, (ii) el usuario experto opera directamente con una versión de la aplicación inteligible para él a la que puede acceder para su mantenimiento y refino en función de la experiencia. Además, al separar el modelo declarativo del dominio de los métodos que lo utilizan se aumenta la reusabilidad de los métodos.

## 2.2.4 Instrumentaciones de KADS

El modelo conceptual de KADS [Wielinga et al., 92], descrito en el apartado 2.1.4, en el que se propone la identificación y organización del conocimiento de un sistema en cuatro niveles como parte de la etapa de análisis de un sistema basado en el conocimiento, propone además una metodología de diseño de estos sistemas, análoga a las que ya existían para los sistemas de información, que tiene en cuenta las características propias de los problemas abordados con la Inteligencia Artificial.

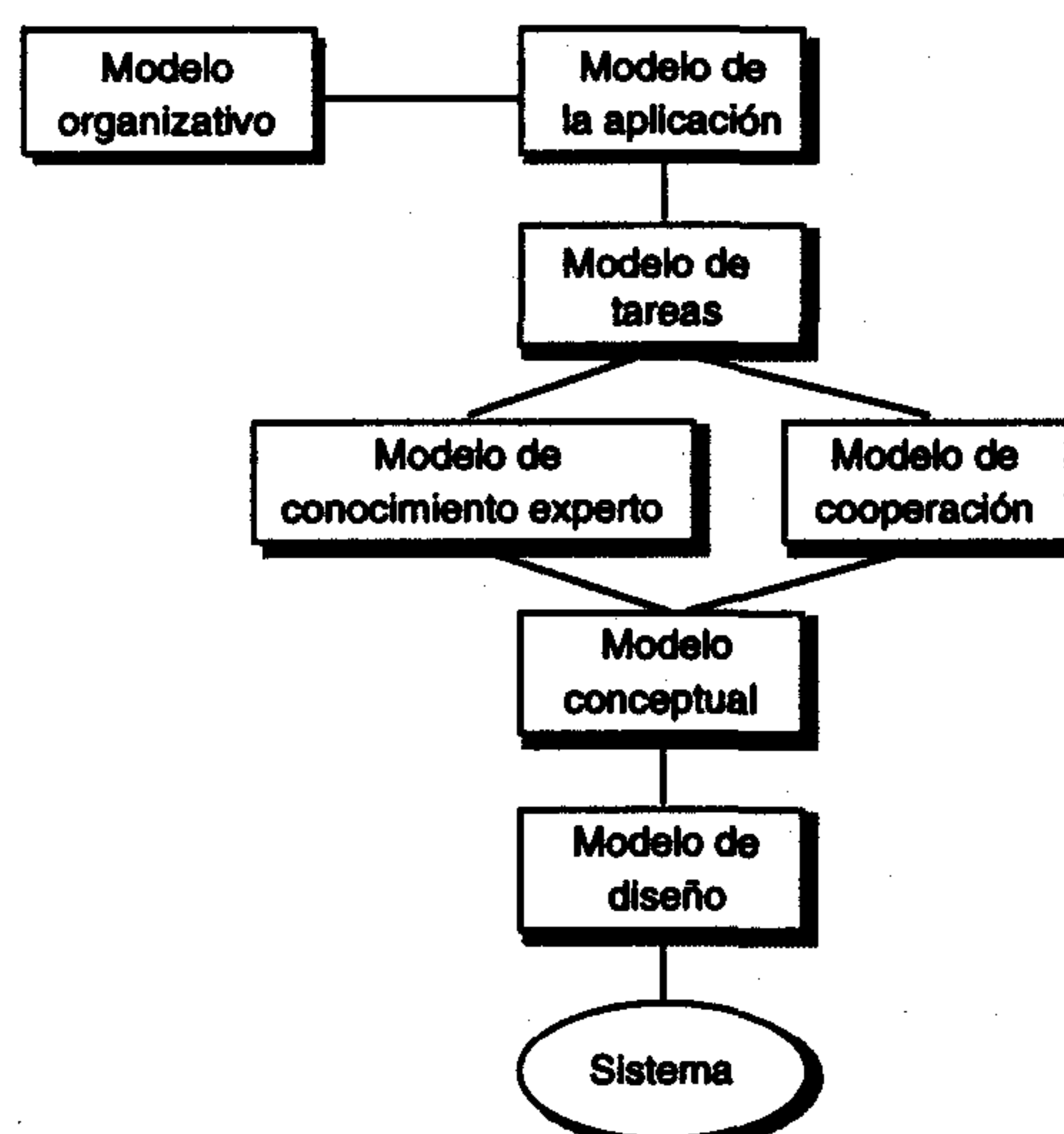
Esta metodología plantea un proceso de ingeniería del conocimiento, con el que acercarse al desarrollo de sistemas basados en el conocimiento, basado en la definición de una serie de modelos que en conjunto proporcionan una visión integral del sistema en el contexto en el que va a utilizarse (ver figura 2.10):

- *el modelo organizativo u operacional*, que describe los aspectos organizativos del sistema en el entorno en el que se va a implantar;
- *el modelo de la aplicación*, que define la funcionalidad del sistema dentro de la organización;
- *el modelo de tareas*, que describe cómo alcanzar la funcionalidad expresada en el modelo de la aplicación mediante la organización jerárquica de las tareas que han de realizarse y la distribución de actividades entre el sistema y su operador;



- *el modelo de conocimiento experto*, que especifica el conocimiento que necesitan las tareas identificadas en el modelo de tareas y el modo en el que ese conocimiento va a emplearse para conseguir del sistema el comportamiento esperado;
- *el modelo de cooperación*, que contiene una especificación de la funcionalidad de las tareas del modelo de tareas que requieren algún nivel de cooperación, como tareas de adquisición de datos durante la resolución de un problema o tareas de explicación;
- *el modelo conceptual*, que describe un modelo de estructuración del conocimiento del experto lo más próximo posible al que éste emplea en la resolución del problema;
- *el modelo de diseño*, que describe los formalismos de representación del conocimiento y las técnicas de inferencia utilizadas para implementar el modelo conceptual.

La construcción de modelos conceptuales y prototipos basados en la propuesta de KADS se ha llevado a cabo con lenguajes como MODEL-K [Karbach, Voß, 93], OMOS [Linster, 93], y KARL [Fensel et al., 91].



*Figura 2.10: Descomposición del proceso de ingeniería del conocimiento  
(adaptada de [Wielinga et al., 92])*

MODEL-K es un lenguaje para modelizar y operacionalizar modelos de experiencia de KADS, que introduce primitivas de modelización específicas para KADS. Ambos modelos, el conceptual y el operacional, se describen a tres niveles: de dominio, de inferencia y de control. En el modelo conceptual, el nivel de dominio describe la estructura del dominio en términos de conceptos y relaciones; el nivel de inferencia determina la estructura de inferencia en términos de fuentes de conocimiento, es decir, operadores básicos y metaclases; y el nivel de control especifica la estructura de control en términos de tareas. En la versión operacional de un modelo conceptual los conceptos se complementan con sus instancias, las relaciones con sus tuplas, las fuentes de conocimiento con sus piezas de código ejecutable, y las tareas con sus flujos de control (ver figura 2.11). El lenguaje MODEL-K incorpora las ventajas de los diseños desarrollados preservando la estructura de conocimiento del sistema: facilita la modificación y extensión del sistema final y da soporte a la generación de explicaciones comprensibles por los usuarios.

MODEL-K es descrito con más detalle en el apartado 2.3.2 ya que, frente a las demás propuestas, permite implementar el nivel estratégico de conocimiento de KADS, lo cual le confiere capacidad de razonamiento reflexivo e introduce cierto grado de dinamismo en la organización del conocimiento del sistema.

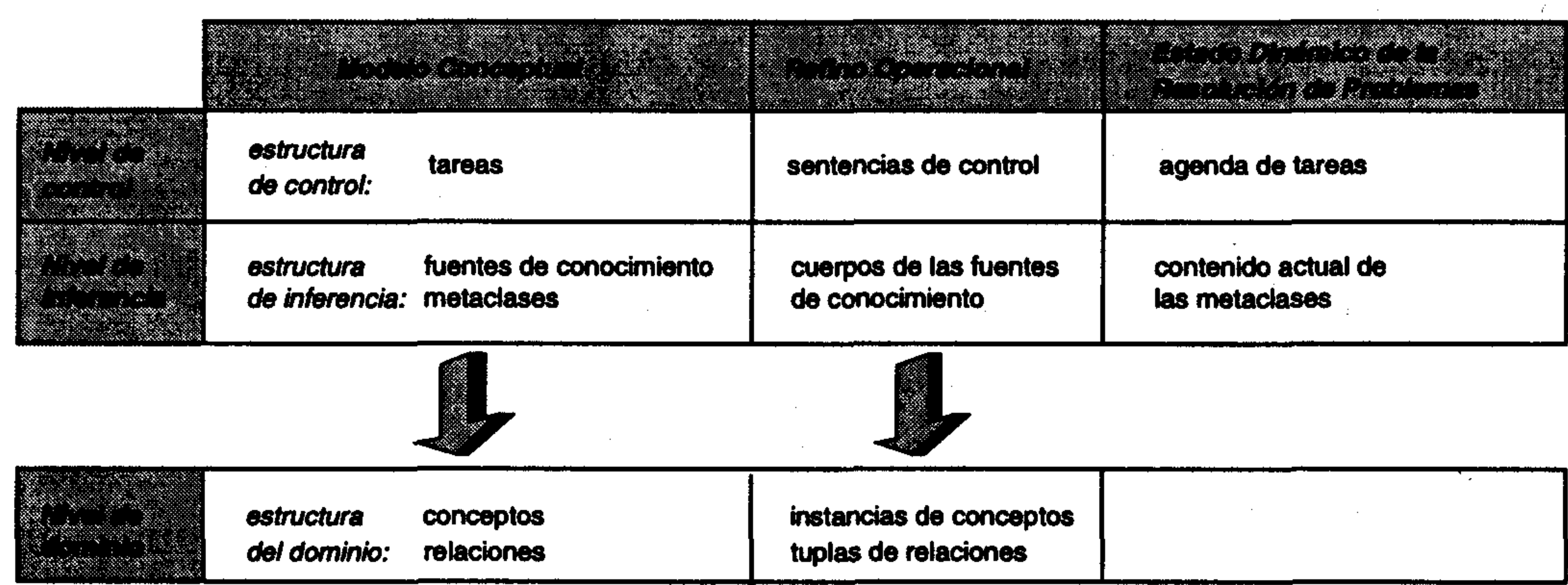


Figura 2.11: El marco de definición de MODEL-K  
(adaptada de [Karch, Vof, 93])

OMOS es otro lenguaje de operacionalización de modelos conceptuales de KADS, los cuales se describen desde dos perspectivas: (i) una basada en el modelo de resolución de problemas, y (ii) otra basada en la estructura del dominio; y cada una de ellas con sus propias primitivas de estructuración del



conocimiento. La primera está descrita en el modelo del método, en el que se representan los cambios y estados observados en el mundo real. Un método consiste en un conjunto de acciones de inferencia, comparables a las fuentes e conocimiento de KADS, un conjunto de roles similares a las metaclases de KADS y una estructura de control comparable a las estructuras de tareas de KADS. La segunda perspectiva se describe en el modelo de dominio, en el que las estructuras del dominio se definen como jerarquías de marcos para los que pueden definirse relaciones ente instancias de diferentes clases. El modelo de tarea de OMOS se construye combinando el modelo de dominio con el modelo de método una vez desarrolladas las estructuras de estos dos modelos. Este modelo de tarea constituye un marco para la adquisición del conocimiento detallado del dominio.

Por otro lado, el lenguaje KARL (*Knowledge Acquisition and Representation Language*) se encuadra dentro de una herramienta más amplia denominado MIKE [Angele et al., 92], que se presenta como un entorno para ayuda al análisis de sistemas a desarrollar. Este entorno cuenta con dos medios:

- Un interfaz hipermedia, que presenta una representación semiformal del modelo conceptual tipo KADS mediante una red cuyos nodos contienen información y que se construye con primitivas para representar actividades de procesos de resolución de problemas, conceptos o entidades relevantes en el proceso de resolución de problemas, flujos de información y flujos de control. Esta estructura define lo que se denomina el hipermodelo, que servirá de punto de partida para la construcción del modelo conceptual, incluyendo facilidades de documentación y explicación.
- El lenguaje para formalizar y operacionalizar el modelo conceptual de la estructura anterior. Este modelo se puede traducir a una versión ejecutable, lo que permite ensayar a nivel de prototipo el modelo realizado, aunque no llega a nunca a obtenerse de el sistema final.

Las propuestas presentadas van dirigidas a facilitar la etapa de análisis del conocimiento de sistemas cuyos modelos conceptuales están inspirados por la metodología KADS; pero, las etapas posteriores de diseño e implementación deben realizarse con otros medios. Este planteamiento es distinto al de las propuestas KREST y PROTÉGÉ-II, descritas a continuación, ya que aunque



comparten características en la forma de modelizar, se diferencian en que persiguen la construcción del sistema final mediante composición de piezas reutilizables.

## 2.3 Modelos Dinámicos de Conocimiento

En los enfoques presentados anteriormente, la selección del modo más adecuado de resolver un problema se tomaba en la etapa de diseño y una vez decidido el método, el sistema se construía a partir de asociaciones fijas tarea-método. Sin embargo, estos modelos, limitan su potencial reusabilidad a su aplicación sobre dominios con características muy similares a las del dominio para el que el modelo fue originalmente diseñado.

La generalización de estos modelos pasa por la flexibilización de los mismos mediante la introducción de formas alternativas de resolver una misma clase de problemas. El diseño de sistemas inteligentes que razonan sobre múltiples métodos de resolución de problemas y que utilizan criterios para seleccionar los métodos más adecuados en una situación concreta supone la asociación de capacidades reflexivas como parte del conocimiento del sistema. Un sistema reflexivo tiene conocimiento sobre su propia capacidad de resolución de problemas y por tanto puede razonar sobre cómo debe enfrentarse a un problema concreto.

La instrumentación de esta capacidad de un sistema para influir en su propio comportamiento puede realizarse mediante una arquitectura en la que se diferencian claramente dos niveles de razonamiento: un nivel de objetos y un metanivel. El *nivel de objetos* incluye todos los módulos de organización e inferencia sobre el conocimiento del dominio que definen la actividad del sistema, y que dependiendo de la metodología de modelización empleada pueden ser métodos, tareas, subtareas, mecanismos, fuentes de conocimiento, etc.; y el *metanivel* que incluye el conocimiento sobre cómo seleccionar dinámicamente los métodos de resolución de problemas más adecuados en un determinado contexto de operación.

Esta sección presenta algunas de las soluciones más significativas dirigidas a flexibilizar el razonamiento de sistemas basados en el conocimiento mediante

la introducción en los mismos de la capacidad de reflexionar sobre el modo de resolver problemas.

### 2.3.1 TIPS

TIPS (*Task Integrated Problem Solving*) [Punch, Chandrasekaran, 93] es un lenguaje específico para problemas de diagnóstico que permite realizar un control flexible de modelos de conocimiento basados en tareas genéricas. En este sentido, TIPS se diseñó para dar respuesta al problema de la integración dinámica de métodos, planteada como posibilidad en el modelo de tareas genéricas de Chandrasekaran [Chandrasekaran, 83, 86, 87], desde un enfoque de simplicidad al proporcionar sólo los mecanismos suficientes para permitir la monitorización de las tareas (objetivos en los términos de TIPS) y una asociación de las tareas con los métodos que pueden realizarlas.

El modelo de organización del conocimiento de TIPS consiste en el árbol de tareas-subtareas-métodos asociado a la metodología de modelización basada en tareas genéricas. La tarea de más alto nivel se descompone en subtareas, que a su vez se descomponen en otras subtareas hasta llegar al nivel de las tareas básicas que son realizadas por métodos básicos. La base para la representación del control usada en TIPS es el sistema *espónsor-selector*. Este mecanismo consiste en una jerarquía de tres partes: un selector, algunos espónsores asociados al selector, y un método asociado a cada espónsor. Los métodos disponibles se agrupan bajo un selector el cual elige uno en función de las medidas de adecuación ofrecidas por los espónsores asociados a los métodos. El mecanismo se activa cada vez que, durante el proceso de resolución de problemas, se llega a un punto a partir del cual hay formas alternativas de alcanzar una respuesta.

Cada espónsor contiene información sobre las condiciones bajo las cuales la aplicación de su método asociado es adecuada. El conocimiento del espónsor se representa en una tabla en la que en cada columna, excepto en la última, hay una pregunta sobre el estado de resolución de problemas del sistema. Cada fila define un patrón de respuestas para cada pregunta e incluye en la última posición un valor cuantitativo sobre el grado de adecuación del método en el contexto descrito por las respuestas. El tipo de preguntas consideradas en la formulación de una tabla se refiere a dos tipos de información (ver figura 2.12):



- *información dinámica de métodos*, en la que se consulta qué métodos se han ejecutado y cuándo;
- *información dinámica de tareas*, en la que se pregunta sobre la realización de tareas u objetivos y cuya respuesta tiene que ser proporcionada por el operador.

El enfoque usado en TIPS es similar al que utiliza [Benjamins, 93] aunque este último sólo permite hacer preguntas locales al método. TIPS permite hacer preguntas con carácter estratégico que hacen depender la aplicabilidad de un método de la disponibilidad de otros métodos, lo cual puede crear problemas si se quieren configurar nuevos sistemas de diagnóstico añadiendo nuevos métodos.

(1) Se ha aplicado ya el método de clasificación?  
(2) El estado actual del diagnóstico es completo?

Pregunta (1)	Pregunta (2)	Valor de plausibilidad
F	?	3
V	F	1
V	V	-3

Figura 2.12: Patrón de equiparación de un espónsor  
(adaptado de [Punch, Chandrasekaran, 93])

Los espónsores, con sus métodos asociados, están agrupados bajo selectores, los cuales determinan qué métodos se van a considerar en cada paso de evaluación y de entre ellos eligen uno. El conocimiento que emplea un selector para elegir un método puede codificarse de tres formas distintas (ver figura 2.13):

- *medidas de adecuación*. La selección se realiza fijándose en las medidas de adecuación proporcionadas por los espónsores. En general se elige aquel método con la medida más alta, pero en caso de empate entre métodos se hace una elección al azar entre ellos o se aplica la lista de prioridades, si es que está definida.



- *lista de prioridades*. Es una lista de métodos que especifica qué método debería elegirse en una situación de empate.
- *patrón de equiparación*. Es una estructura similar a la de los espónsores, pero en este caso, la última columna incluye nombres de métodos en vez de grados de adecuación. Se utiliza en ocasiones especiales para saltarse el mecanismo normal de elección.

Entonces, la arquitectura TIPS está organizada a dos niveles:

- un nivel de decisiones locales sobre adecuación de métodos codificado en el espónsor en términos de conocimiento sobre el progreso de la resolución del problema y la historia de la activación de métodos, es decir, sobre la información con respecto al éxito/fracaso obtenido y a la selección de métodos efectuada previamente;
- un nivel de decisiones globales que definen la elección del siguiente método a aplicar y que está codificado en los selectores.

<i>Espónsore asociados</i>	Espónsor de clasificación jerárquica Espónsor de selección de síntomas Espónsor de selección de hipótesis		
<i>Lista ordenada de prioridades</i>	(Síntomas, Jerarquía, Hipótesis)		
<i>Patrón de equiparación</i>	(1) Se ha aplicado ya el método de clasificación? (2) Se ha examinado la jerarquía completamente?		
	Pregunta (1)	Pregunta (2)	Método seleccionado
	No	No	clasificación jerárquica
	Si	No	síntomas

Figura 2.13: Conocimiento de un selector  
(adaptado de [Punch, Chandrasekaran, 93])

El ciclo de razonamiento de una estructura espónsor-selector se da por terminado con la respuesta de dos espónsores distinguidos del resto: *return* y *fail*. Estos indican si la tarea asociada al selector se ha realizado con éxito o no, y normalmente tienen mayor prioridad que los demás espónsores de un selector.



Un aspecto problemático del mecanismo espónsor-selector de TIPS es que la jerarquía de objetivos de la estructura de tareas está representada sólo implícitamente, con lo que se empaña la claridad del modelo de razonamiento. Por otra parte, no hay una distinción clara entre tarea y método a los ojos del selector ya que un selector puede tener que elegir en un momento dado entre métodos asociados a distintos tipos de objetivos. Por tanto, en un paso de evaluación, en el que se ve qué hacer a continuación, pueden examinarse tanto métodos como tareas, por lo que pueden obtenerse dos resultados: o bien se selecciona un método para alcanzar el objetivo de la tarea, o bien se selecciona un nuevo subobjetivo a alcanzar.

Entonces, en TIPS se hacen dos tipos de razonamiento metanivel:

- (i) seleccionar el método adecuado para una tarea, y
- (ii) una vez elegido un método, determinar dinámicamente el control sobre sus subtareas, esto es, determinar cuál es el siguiente objetivo que hay que conseguir.

Además, aunque en TIPS cada tarea tiene su propia estrategia de control, que es la estructura espónsor-selector, ésta es una estructura de control fija que no puede ni extenderse, ni modificarse. Así, no es posible definir estrategias que iteren sobre los métodos disponibles en caso de que un método falle, o que invoquen una tarea en caso de que el mejor método, el más apropiado, no pueda aplicarse.

Por último, TIPS no tiene criterios de aplicabilidad, es decir, estos criterios suelen estar embebidos en los criterios de adecuación. Así, en ocasiones no queda claro porqué no se selecciona un método, si porque no es aplicable o porque no es adecuado.

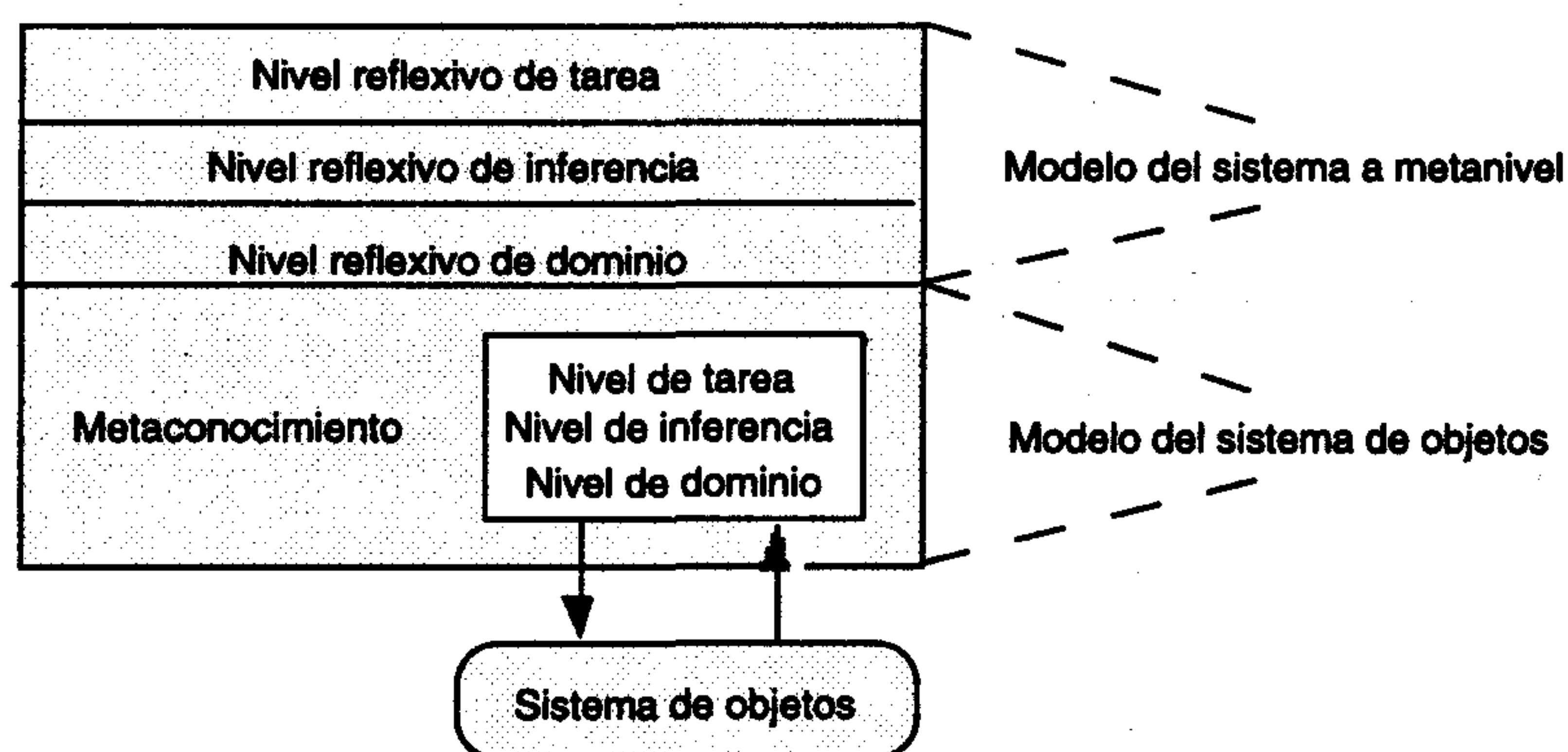
### **2.3.2 MODEL-K**

MODEL-K [Karbach, Voß, 93], como se presentó en el apartado 2.2.2, es un lenguaje de propósito general para modelizar y operacionalizar modelos de experiencia de KADS, que no provee de facilidades específicas para la selección dinámica de métodos, pero que sin embargo, es interesante porque es uno de los pocos lenguajes que implementan modelos KADS que incluyen el nivel

estratégico. El tipo de razonamiento metanivel de MODEL-K contempla aspectos como: análisis de la dificultad y/o posibilidad de solución de un problema, predicción y planificación de los recursos necesarios para resolver un problema, simplificación de problemas, o detección de puntos muertos en un proceso de resolución.

A la vista de estas actividades, que requieren razonar sobre el sistema de objetos subyacente, el nivel estratégico modeliza un sistema de resolución de problemas reflexivo, en la línea de [Fensel, Harmelen, 93], que se contempla como otro módulo de resolución de problemas a metanivel que puede ser modelizado con el mismo esquema de tres niveles (dominio, inferencia y control) usado para las actividades definidas al nivel de objetos. Los niveles de inferencia y control incluyen la descripción de las actividades enumeradas anteriormente, y el control sobre estas actividades lo realiza un planificador. El nivel de dominio contiene un modelo del sistema de objetos sobre los que hay que razonar y algún meta-conocimiento, como por ejemplo, consistencia de datos en el sistema de objetos o condiciones necesarias para resolver un problema (ver figura 2.14).

Una de las ventajas de MODEL-K es que es un lenguaje abierto que soporta cualquier número de actividades metanivel que al estar representadas explícitamente, pueden añadirse o modificarse sin dificultad, y asociarse a casi todas las tareas del nivel de objetos para controlar su ejecución.



*Figura 2.14: El nivel estratégico en MODEL-K  
(adaptada de [Karch, Voß, 93])*

Por otra parte, aunque el planificador de actividades metanivel se formula con un lenguaje procedural consistente en secuencias, alternativas y bucles, lo cual



le permitiría definir estrategias de control flexibles, en realidad estas estrategias son poco flexibles y tienen un acceso limitado a las tareas. Son rígidas porque el planificador también es rígido (impone una secuencia de tareas fija), en contra de un planificador flexible que es el necesario para conseguir verdaderas arquitecturas cognitivas flexibles este tipo de actividad (con decisiones sobre modos alternativos de resolver problemas). Además, ofrece un acceso limitado a las tareas porque el planificador tiene efecto sólo sobre las tareas de más alto nivel de la estructura de control, por lo que las tareas intermedias no pueden tener asociadas actividades metanivel.

Como desventajas cabe destacar que, al igual que KADS, MODEL-K no distingue entre tareas y métodos, por lo que la flexibilidad que aporta el lenguaje se ve empañada por la dificultad en discernir si una decisión que se está tomando implica un cambio de objetivo (o tarea) o una elección de un método para llevar a cabo el objetivo en curso; de forma similar a como ocurre con TIPS.

### 2.3.3 TroTelC

[Vanwelkenhuysen, Rademakers, 90, 93] plantearon un marco computacional en el que operacionalizar modelos de conocimiento definidos con la metodología de los Componentes de la Experiencia de Steels [Steels, 90] (ver apartado 2.1.5), que ilustraron con su aplicación en el sistema TroTelC. TroTelC es un sistema experto que realiza diagnóstico de problemas en circuitos digitales impresos, si es que en el proceso de fabricación de dichos circuitos se detectan anomalías no asociadas a defectos de fabricación. TroTelC identifica qué funciones no se están realizando correctamente en el circuito y señala qué componentes físicos son los responsables de ese funcionamiento.

En el modelo empleado en TroTelC se representan explícitamente las tareas, métodos y modelos del dominio que emplea el sistema. La singularidad de esta propuesta de [Vanwelkenhuysen, Rademakers, 90, 93] es que admite la realización de una tarea por diferentes métodos de resolución de problemas. Los métodos se clasifican en dos tipos:

- (i) *métodos de acción*, que se corresponden con primitivas de inferencia básicas, y



- (ii) *métodos de descomposición de tareas*. Ambos tipos de métodos se caracterizan, entre otras cosas, por una condición de aplicabilidad que indica cuándo el método es apropiado para realizar una tarea y determina su selección.

```
(defconcept DIFFERENTIATE-BETWEEN-SUSPECTS
  (a task
    (problem-solving-methods
      [method-list
        (an eliminate-by-implication
          (condition [condition (?context)
                     (availability-of-test-results ?context)]))
        (an eliminate-by-testing
          (condition [condition (?context)
                     (current-suspect-has-associated-test ?context)]))]))))

(defconcept ELIMINATE-BY-IMPLICATION
  (a problem-solving-method
    (task-decomposition
      [list-of-tasks (remove-inputs-to-working-components-from-set-of-suspects
                     remove-outputs-of-failing-components-from-set-of-suspects)])
    (control-structure sequential)))

(defconcept ELIMINATE-BY-TESTING
  (a problem-solving-method
    (task-decomposition
      [list-of-tasks (take-suspects collect-tests propose-tests-to-execute
                     execute-tests deduce-facts-from-test-results
                     reduce-set-of-suspects)])
    (control-structure sequential)))
```

Figura 2.15: Especificación de tareas y métodos en TroTelC  
(de [Vanwelkenhuysen, Rademakers, 90])

En el caso de TroTelC, la tarea de diagnóstico se lleva a cabo con el método *cubrir & diferenciar*, descompuesto en las subtareas *cubrir síntomas* y *discriminar entre sospechosos*. Para la segunda subtask se dispone de dos métodos: *eliminar por implicación* y *eliminar por prueba*. Sus condiciones de aplicabilidad son respectivamente: "disponibilidad de resultados de tests" y "existencia de test para el sospechoso actual" (ver figura 2.15). Estas condiciones representan sólo requisitos necesarios para usar los métodos, pero la selección final de un método entre varios candidatos está implícita ya que no se especifica ningún tipo de conocimiento que pudiese dirigir la elección, como prioridades entre métodos o criterios de adecuación.



TroTelC dispone de una estrategia de control implícita y sencilla que, al contrario de lo que sucede en MODEL-K, puede aplicarse sobre todas las tareas del modelo de conocimiento. Sin embargo, como limitación tiene que no permite añadir más actividades metanivel a la única disponible, esto es, selección de métodos. Además, la estrategia de control de esta actividad metanivel no es accesible ni modificable, por ejemplo, para hacer posible una iteración entre métodos aplicables en el caso de que fallase el seleccionado previamente.

## 2.4 Conclusiones

En este capítulo se ha mostrado la evolución de los sistemas basados en modelos estructurados de conocimiento desde la perspectiva de su capacidad de interacción con los usuarios.

En primer lugar se presentaron las propuestas de modelización del conocimiento más significativas: agente de Newell, tareas genéricas, métodos *role limiting*, niveles de conocimiento de KADS, componentes de la experiencia y unidades cognitivas. Estas propuestas constituyen los primeros pasos para mejorar la interacción entre personas y sistemas al buscar modos de representar y organizar el conocimiento de una persona que se alejan de los aspectos de implementación del sistema que ha de hacer uso de ese conocimiento, dándole la posibilidad al usuario de participar en la etapa de análisis de la aplicación y consiguiendo como valor añadido un mayor nivel de usabilidad en el sistema final.

A continuación se presentaron las soluciones instrumentales de modelos desarrollados a partir de alguna de las metodologías de modelización anteriores: KREST, PROTÉGÉ-II, KSM e instrumentaciones KADS. Las soluciones presentadas en este sentido se caracterizan por dos cosas: (i) entienden que la construcción de un sistema basado en el conocimiento es el resultado de ensamblar una colección de piezas ya programadas contenidas en una biblioteca y, (ii) manejan modelos de resolución de problemas únicos al asociar a cada tarea un sólo método con el que realizarla.



Finalmente, tras observar que la reusabilidad de los modelos construidos con las herramientas de modelización anteriores depende de que se alcance un alto nivel de ajuste entre las características del dominio problema en el que se desarrolló el modelo y las del dominio objetivo; se describieron las propuestas encaminadas a flexibilizar los modelos de conocimiento dotando al modelo de capacidad reflexiva mediante la cual puede elegir dinámicamente qué método de resolución de problemas debe emplearse entre un conjunto de métodos alternativos.

En este punto cabe decir que las soluciones de interacción que ofrecen los sistemas inteligentes han evolucionado significativamente para acercar los modelos de comprensión de usuarios y sistemas, pero se han olvidado de los aspectos de presentación por lo que toda la información que es capaz de generar el sistema podría no ser adecuadamente percibida por los diferentes usuarios.

### **3. Interacción Usuario-Sistema Como Resultado De Una Presentación Elaborada**

---

Un factor fundamental para alcanzar el éxito en la comunicación de un usuario con su sistema es la capacidad de este último para expresar adecuadamente la información que quiere transmitir. El concepto de adecuación en este contexto ha de entenderse como la posibilidad de expresar el mensaje en modo atractivo y comprensible para el usuario, lo que además redundará en un incremento de la usabilidad del sistema. Con este objetivo, la investigación en el lenguaje de comunicación hombre-máquina ha estado usualmente dirigida por el análisis de la comunicación humana. Es por ello que, desde el principio de la investigación en interfaces de usuario, se han dedicado numerosos esfuerzos en desarrollar sistemas capaces de comunicarse con su usuario en lenguaje natural basándose en el análisis de conversaciones mantenidas entre personas.

La capacidad de un sistema de expresarse en forma similar a como lo hacen las personas ha sido uno de los aspectos fundamentales y primeros que llevaron a calificar a un interfaz como inteligente, aunque con el tiempo este concepto ha ido incorporando nuevos elementos producto de la evolución de la tecnología informática, como son la coordinación de múltiples medios de comunicación. El aprovechamiento adecuado de estos nuevos dispositivos multimedia originó una nueva línea de investigación orientada a conseguir la adaptación automática del interfaz de un sistema al tipo de usuario con el que está interaccionando y a su modelo de conocimiento. Con la introducción de estos potentes recursos de presentación en los interfaces aumentó la complejidad de los mismos, incrementándose el coste y la dificultad de su diseño y construcción; lo que en los últimos años ha provocado la necesidad de desarrollar modelos abstractos de interfaces como etapa previa a su construcción.



En las secciones siguientes se presenta, en primer lugar, un breve resumen de las características de la primera generación de sistemas con presentaciones en lenguaje natural. Después se describen las soluciones más significativas de generación de presentaciones en lenguaje natural, ya calificadas de inteligentes en la bibliografía, que posteriormente han sido enriquecidas mediante el uso de recursos multimedia. La cualidad de inteligencia de esta nueva generación de sistemas se adquiere por la incorporación de una representación explícita de la intención perseguida por el sistema al comunicarse con el usuario y del efecto buscado en el receptor, es decir, se empiezan a considerar modelos de los usuarios; que también son descritos a continuación. Finalmente, como consecuencia del aumento en la complejidad del diseño y construcción de estos interfaces se describen algunas características de las soluciones de diseño de interfaces de usuario a alto nivel.

### **3.1 Presentaciones de la Primera Generación**

Los primeras soluciones dirigidas a dotar a las aplicaciones de la capacidad de expresar sus conclusiones se basaron en el uso de plantillas predefinidas de texto asociadas con estructuras o procedimientos del código de la aplicación. Estas plantillas disponían de variables que tomaban valor como resultado de la ejecución de algún procedimiento. De esta manera, los desarrolladores de la aplicación anticipaban las preguntas que podían hacer los usuarios y para cada una de ellas, dependiendo de la situación, generaban las respuestas.

Aunque la gran ventaja de este enfoque es su simplicidad, sobre todo cuando el número de situaciones en las que el sistema necesita expresar sus resultados es pequeño y está bien caracterizado, el uso de plantillas predefinidas de texto tiene serias limitaciones entre las que destaca la dificultad en mantener la consistencia entre el comportamiento de sistema y sus presentaciones al poder cambiar el código del sistema de forma independiente del texto de estas plantillas.

Con el objetivo de resolver este problema de falta de fidelidad de las presentaciones obtenidas con plantillas, se construyeron sistemas que generaban sus presentaciones haciendo transformaciones o traducciones directamente del código de los mismos, entre los que se encuentran DTA

(*Digitalis Therapy Advisor*) [Swartout, 77], y MYCIN [Buchanan, Shortliffe, 84]. Así, los usuarios podían confiar en que el sistema estaba haciendo realmente lo que decía y además, los costes de mantenimiento del sistema disminuían al reflejarse automáticamente los cambios del código en las presentaciones. Por otra parte, como la estructura de las presentaciones seguía la estructura del código del sistema, se facilitaban las tareas de construcción y depuración. Sin embargo, este enfoque tiene también importantes deficiencias:

- Incapacidad de producir presentaciones naturales al corresponderse éstas demasiado literalmente a la estructura del código del sistema.
- Inflexibilidad en el tipo de presentaciones que se ofrecen ya que normalmente sólo hay una manera de explicar el conocimiento o comportamiento del sistema.
- Insensibilidad a aspectos del contexto de la consulta ya que este contexto no se representa y cada par pregunta-respuesta se ve de forma independiente.

Entonces, de cara a mejorar la calidad de las presentaciones en lenguaje natural se planteó la necesidad de desarrollar teoría sobre los principios lingüísticos que gobiernan las formas en que los sujetos transmisores de información seleccionan y estructuran la misma para alcanzar el efecto deseado sobre los receptores de esa información. Por tanto, para enriquecer sus presentaciones un sistema debe seleccionar la información relevante que le ayude a alcanzar un propósito comunicativo específico y organizar esa información en una expresión coherente en lenguaje natural, lo que en el área de investigación en lenguaje natural se denomina planificación de texto.

### 3.1.1 Los patrones de la estructura de un discurso

[McKeown, 85], en su sistema TEXT, mostró que algunas colecciones de los predicados retóricos obtenidos del estudio de la naturaleza de las relaciones que existen entre las sentencias o frases de un discurso coherente, como por ejemplo *identificación*, *descripción* y *recomendación*, podían codificarse en estructuras o patrones que podían guiar la generación de textos compuestos de múltiples frases. En el desarrollo de TEXT, McKeown identificó dos tipos de



problemas: (i) por una parte, decidir qué información debe incluirse en la respuesta, y por otra, (ii) organizar esa información en un texto coherente.

Del análisis de textos, McKeown observó que no sólo los predicados retóricos solían combinarse de formas determinadas sino que algunas de estas combinaciones eran más apropiadas que otras dependiendo del propósito del discurso. A estas estructuras de relaciones retóricas las denominó *rethorical schemata* [McKeown et al., 90]. Por ejemplo, tras observar que las personas normalmente definen o describen objetos en términos de sus partes constituyentes, McKeown definió el *schemata* mostrado en la figura 3.1.

PASOS EN LA DESCRIPCIÓN DE UN OBJETO	SCHEMATA
1. Identificar el objeto como miembro de alguna clase genérica	Constituency schema Identification/Attributive
2. Introducir los constituyentes del objeto definido	Constituency
3. Proporcionar información característica de cada constituyente	Cause-effect*/Attributive*/ {Depth-identification/Depth-attributive {Particular-illustration/Evidence} {Comparison; Analogy/Renaming}}+ {Amplification/Explanation/Attributive/Analogy}
4. Proporcionar información propia o analógica del objeto definido	

Figura 3.1: Schemata asociado a la descripción de un objeto ("{" opcionalidad, "/" alternativa, "+" aparición 1 o más veces, "\*" aparición 0 o más veces)

La generación de texto se hace rellenando, de forma secuencial, cada una de las entradas del patrón con un predicado instanciado con alguna proposición de una base de conocimiento en la que está la información a presentar, o bien con un patrón completo cuyo nombre coincida con el de la entrada. Una vez instanciado, la estructura de discurso representada en el patrón guía el proceso de generación de texto.

Como normalmente, el número de alternativas con las que instanciar un patrón es muy grande, se utilizan un conjunto de restricciones que modifican el foco de atención de una frase a otra dentro del discurso. Entonces, cuando hay que hacer una elección sobre qué decir a continuación, las restricciones de foco seleccionan una proposición atendiendo a las siguientes prioridades:

- (i) centrarse en un tema que acaba de presentarse,
- (ii) mantener el tema actual,

- (iii) volver a un tema introducido anteriormente,
- (iv) centrarse en un tema que tenga la mayor relación con los temas ya presentados.

Aunque, a partir de los trabajos de McKeown, hubo muchos investigadores que utilizaron estos patrones de la estructura de discursos para generar texto compuesto de múltiples sentencias, éstos presentan serias limitaciones, como son:

- Falta de una representación explícita de la intención del texto presentado. Esto es un problema serio si el sistema quiere mantener diálogos con el usuario en los que éste puede hacer preguntas derivadas de otras anteriores solicitando una alternativa o una aclaración de la respuesta. Si el sistema no recuerda la intención que había en las frases que presentó entonces no puede determinar con qué parte de la explicación ha tenido problemas el usuario, y por tanto no puede ofrecer alternativas.
- Rigidez para gestionar la ocurrencia de eventos que aparecen de forma oportunista en los textos reales; como por ejemplo, la tendencia a definir un concepto inmediatamente después de haberlo mencionado durante una explicación. La solución de este problema con los patrones obligaría a introducir un número excesivo de predicados retóricos en su estructura que oscurecerían la naturaleza del patrón y añadirían complejidad y coste a su uso.

### 3.1.2 La teoría de Estructuras Retóricas

RST (*Rethorical Structure Theory*) [Mann, Thompson, 87] es una teoría sobre la estructura de cuerpos de texto o discursos en la que se define la conexión entre los predicados retóricos que determinan las frases que componen el texto y las intenciones que subyacen a esas piezas de texto. Cada relación retórica en RST define restricciones sobre la forma de combinar las dos entidades relacionadas, y una especificación del efecto que el que lanza el mensaje espera causar en las creencias del que lo recibe.

El número de estructuras de texto basadas en predicados RST se define por un conjunto de patrones o esquemas que actúan como la gramática de RST. Cada esquema representa un patrón abstracto que consiste en un pequeño número



de porciones de texto y una especificación de las relaciones RST que deben existir entre ellas. Aunque el patrón más común consiste en dos piezas de texto, el núcleo y el satélite, y una única relación entre ellos, estos patrones pueden relacionar más de dos piezas de texto y además de forma recursiva: el texto que sirve como núcleo o satélite en un patrón puede estar descrito por otro patrón que a su vez puede descomponerse en otras piezas de texto, etc.

RELACIÓN RST	OPERADOR RST
RELATION NAME: Circumstance CONSTRAINTS ON N (nucleus): None CONSTRAINTS ON S (satellite): Presents a situation (not unrealized) CONSTRAINTS ON N + S COMBINATION: S sets a framework in the subject matter within which the hearer is intended to interpret the situation presented in N EFFECT: The hearer recognizes that the situation presented in S provides the framework for interpreting N.	Name: Circumstance Results: ((BMB SPEAKER HEARER (CIRCUMSTANCE-OF ?X ?CIRC))) Nucleus + Satellite requirements/subgoals: ((OR (BMB SPEAKER HEARER (HEADING.R ?X ?CIRC)) (BMB SPEAKER HEARER (TIME.R ?X ?CIRC)))) Nucleus requirements/subgoals: ((BMB SPEAKER HEARER (TOPIC ?X))) Nucleus growth points: ((BMB SPEAKER HEARER (ATTRIBUTE-OF ?X ?ATT))) Satellite requirements: ((BMB SPEAKER HEARER (TOPIC ?CIRC))) Satellite growth points: ((BMB SPEAKER HEARER (ATTRIBUTE-OF ?CIRC ?VAL))) Order: (NUCLEUS SATELLITE) Relation phrases: (" ")

Figura 3.2: Ejemplo de relación y operador RST asociado (de [Hovy, 91])

Aunque RST fué diseñado fundamentalmente para investigar aspectos lingüísticos desde el análisis de textos, [Hovy, 88, 91] presentó un modelo formal de generación de texto a partir de relaciones RST basado en la asociación de operadores de planificación a estas relaciones. Hovy emplea un mecanismo de planificación de arriba a abajo que estructura un conjunto de piezas de texto recibidas como entrada en un discurso coherente. La definición de operadores a partir de relaciones RST se hace asociando el efecto esperado de la relación con el atributo *resultados* del operador, cuyo valor se interpreta como el objetivo de la comunicación que se alcanza utilizando una estrategia definida por el nombre de la relación, y donde las restricciones sobre el núcleo



y satélite de una relación se convierten en subobjetivos del formalismo de Hovy.

La figura 3.2 muestra un ejemplo de relación RST y el operador RST asociado. Este operador dice que para poder alcanzar el estado en el que el que habla cree que tanto él como el oyente creen que ?CIRC es una circunstancia de algún evento ?X, el que habla debe establecer ?X y la información circunstancial ?CIRC a continuación (mediante la solución de los subobjetivos (BMB SPEAKER HEARER (TOPIC ?X)) y (BMB SPEAKER HEARER (TOPIC ?CIRC)) respectivamente). Las restricciones sobre la combinación del núcleo más el satélite analizan que cualquiera que sea el valor ligado a la variable ?CIRC estará relacionado con el evento ligado a ?X por una relación *HEADING.R* o *TIME.R*. Un ejemplo del texto que podría generarse utilizando este operador sería: *Tom está en ruta a Sasebo. Tom va en dirección Sur-Suroeste*; en la que la primera frase es el núcleo y la segunda el satélite, que mantiene una relación de *Circumstance* con el núcleo.

[Hovy, 88, 91] ofrece una primera aproximación al diseño de sistemas capaces de participar en diálogos con el usuario al definir enlaces explícitos entre relaciones retóricas e intenciones. Sin embargo, este modelo no resulta adecuado en numerosas ocasiones ya que en la realidad las asociaciones entre relaciones e intenciones no son una-a-una, sino varias-a-varias, y ésto no puede representarse en el formalismo de Hovy.

## 3.2 Presentaciones Inteligentes en Lenguaje Natural

Como consecuencia natural del empeño en aplicar el lenguaje natural como medio de comunicación de un sistema con sus usuarios, surge una nueva generación de sistemas con capacidad para generar presentaciones, ya denominadas inteligentes, que se generan por la incorporación de una representación explícita de la intención perseguida por el sistema al comunicarse con el usuario y del efecto buscado en el receptor. En relación con este mismo aspecto, se empiezan a considerar soluciones de modelización de características de los usuarios de estos sistemas que contribuyan a favorecer el éxito de estas presentaciones al conjugar las intenciones comunicativas del



sistema con las necesidades de los usuarios; es decir, se empiezan a aplicar modelos de usuarios, que son descritos en la sección 3.4.

### 3.2.1 El modelo de planificación de Moore y Paris

En línea con los trabajos de Hovy sobre RST, [Moore, Paris, 93], [Moore, 95] presentan un lenguaje de planificación que capacita a un sistema para construir planes de explicaciones que recogen el conocimiento necesario para hacer posible su participación en un diálogo con el usuario. Partiendo del objetivo que define la intención del sistema con respecto a las modificaciones que quiere introducir en el conocimiento del usuario, el planificador busca los recursos lingüísticos necesarios con los que alcanzar ese objetivo. Estos recursos pueden ser actos de habla, que son directamente satisfacibles, o bien estrategias retóricas que indican qué se puede decir a continuación y cómo ésto se relaciona con lo que se ha dicho antes. Para poder capturar de forma explícita las intenciones y la estructura retórica de las presentaciones se distinguen dos tipos de objetivos:

- *objetivos comunicativos*, que representan la intención del sistema de afectar a las creencias del usuario. Por sí mismos no causan la generación directa de ningún texto.
- *objetivos lingüísticos*, que se corresponden con los medios lingüísticos disponibles para alcanzar los objetivos comunicativos. Estos sí generan texto y pueden ser de dos tipos: *actos de habla* y *objetivos retóricos*. Los actos de habla son cuatro: Informar, Recomendar, Solicitar, Ordenar; pueden ser directamente asociados a frases de texto que formarán parte de la explicación final, y se consideran primitivas del planificador. Los objetivos retóricos, como Motivación o Circunstancia, no pueden alcanzarse directamente y son descompuestos en uno o más subobjetivos, por lo que acaban generando otros objetivos comunicativos y lingüísticos. Al final, las hojas del árbol que describe el plan del texto a generar incluirán sólo actos de habla.

Con esta diferenciación entre objetivos se consigue: (i) por una parte, representar la relación varias-a-varias entre intenciones y estrategias retóricas que pueden alcanzarlas, y (ii) por otra parte, tener contenida en la estructura completa de un plan la representación explícita de las intenciones del sistema y

el registro de los actos de habla y estrategias retóricas requeridas para conseguirlas.

La estructura de los operadores de planificación utilizados incluye los siguientes elementos:

- un *efecto*, que es la caracterización del objetivo que puede alcanzarse con el uso del operador.
- una *lista de restricciones*, que define las condiciones que han de satisfacerse para poder conseguir del operador el efecto deseado y que pueden referirse a hechos de alguna base de conocimiento, información del modelo de usuario, información de la historia del diálogo, o información sobre el plan en el que se aplica.
- un *núcleo*, que define el subobjetivo más importante de cara a alcanzar el efecto del operador.
- uno o más *satélites*, que definen subobjetivos adicionales que contribuyen a alcanzar el efecto deseado.

La figura 3.3 muestra un ejemplo de operador y su significado.

OPERADOR	SIGNIFICADO
<p><i>EFFECT:</i></p> <p>(PERSUADED ?hearer (DO ?hearer ?act))</p> <p><i>CONSTRAINTS:</i></p> <p>(AND (STEP ?act ?goal)</p> <p>(GOAL ?hearer ?goal)</p> <p>(MOST-SPECIFIC ?goal))</p> <p><i>NUCLEUS:</i></p> <p>(FORALL ?goal</p> <p>(MOTIVATION ?act ?goal))</p> <p><i>SATELLITES:</i></p> <p>nil</p>	<p>Para alcanzar el estado en el que usuario está persuadido de realizar un acto,</p> <p><b>SI</b></p> <p>el acto es un paso para alcanzar algún objetivo/os del usuario, y</p> <p>el objetivo/os es el más específico con respecto a cualquier opción de refino</p> <p><b>ENTONCES</b></p> <p>motivar el acto en términos del objetivo/os</p>

Figura 3.3: Operador de planificación para convencer al usuario de que realice un acto (de [Moore, 95])

Además, como se ha mencionado antes, el modelo de Moore y Paris incorpora una representación del conocimiento, aplicable tanto para modelizar el conocimiento del usuario como el del sistema, expresado mediante una serie de



predicados con los que se expresa lo que el usuario sabe, cree, o es capaz de hacer.

El resultado de la planificación es una estructura de una explicación o cuerpo de texto con forma de árbol en la que los nodos representan objetivos a varios niveles de abstracción, siendo la raíz el objetivo inicial, y las hojas representan actos comunicativos primitivos. Esta estructura de discurso se pasa finalmente a un generador de lenguaje natural que es el que elabora la información de pantalla.

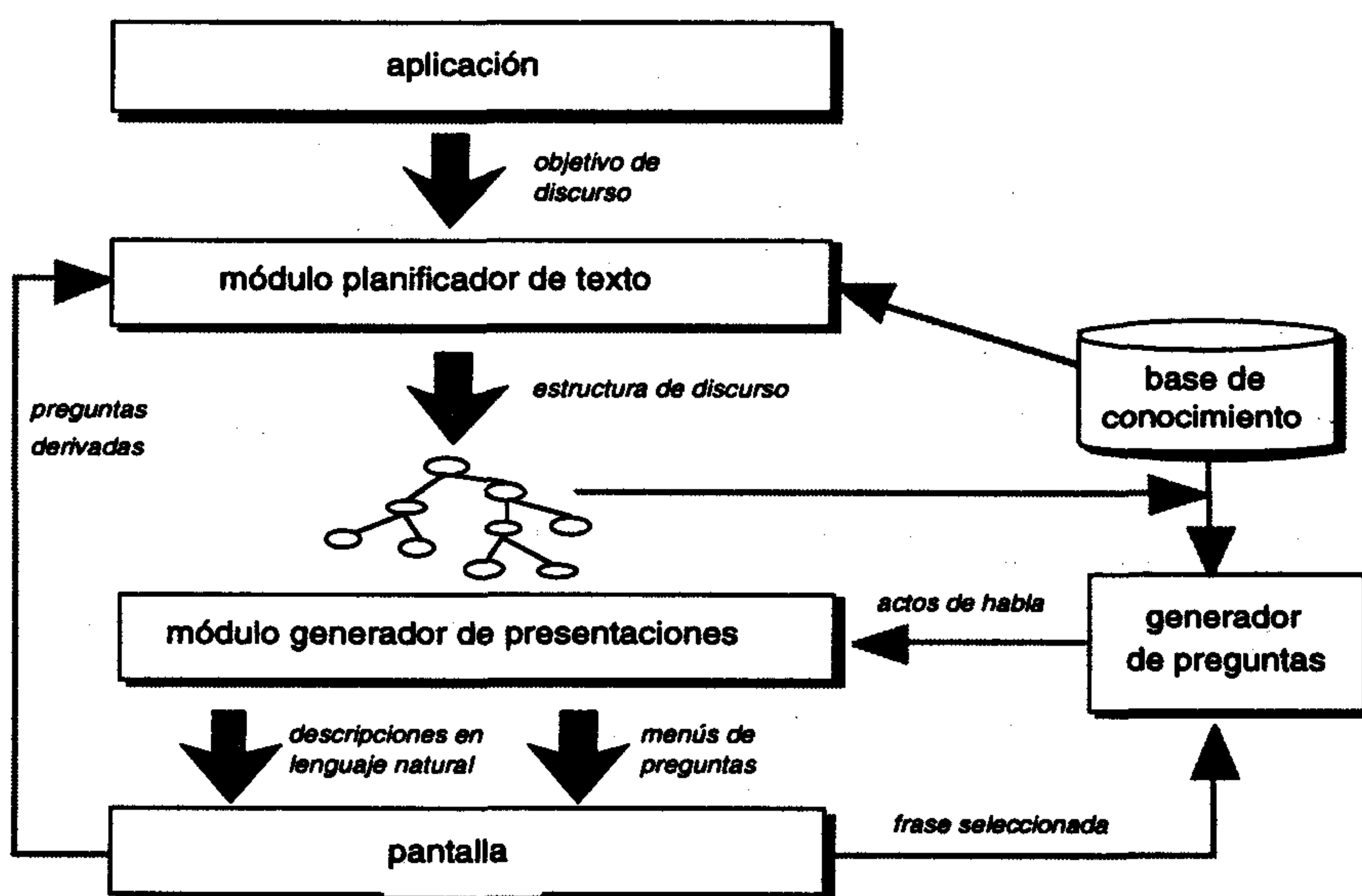


Figura 3.4: Estructura del sistema generador de preguntas

El planificador de Moore y Paris, fue desarrollado inicialmente para servir de base al componente explicativo de un sistema experto interactivo capaz de generar dinámicamente preguntas derivadas de la información presentada en pantalla [Mittal, Moore, 95], [Moore, 95] (ver figura 3.4). Ofrece otra solución de operacionalización de operadores RST que extiende las posibilidades de los operadores del modelo de Hovy (ver apartado 3.1.2) con la representación de las relaciones varias-a-varias entre intenciones y estrategias comunicativas con las que alcanzarlas, a la vez que mantiene las ventajas de eficiencia computacional del uso de patrones de predicados retóricos sin renunciar a la representación de las intenciones que hay detrás de cada pieza de texto presentado. Este planificador fue posteriormente incorporado y generalizado para generar presentaciones multimedia en el sistema WIP [Wahlster et al., 93] (ver apartado 3.3.2).

### 3.2.2 La teoría de actos comunicativos de Maybury

[Maybury, 92] plantea una solución al problema de la falta de naturalidad y rigidez de las explicaciones en los sistemas tradicionales introduciendo una representación explícita de la estructura de la presentación que puede emplearse para generar diferentes formas textuales de mostrar información.

Su propuesta está basada en una teoría de actos comunicativos construida desde el análisis de las explicaciones humanas para poder identificar y transferir el conocimiento sobre la forma en que éstas se generan a los sistemas. Guiado por este análisis, elabora una clasificación de las frases o sentencias componentes de una explicación a partir de su contenido y función comunicativa que incluye cuatro categorías: Descripción, Narración, Exposición y Argumento; y algunos subtipos particulares dentro de estas clases (ver figura 3.5).

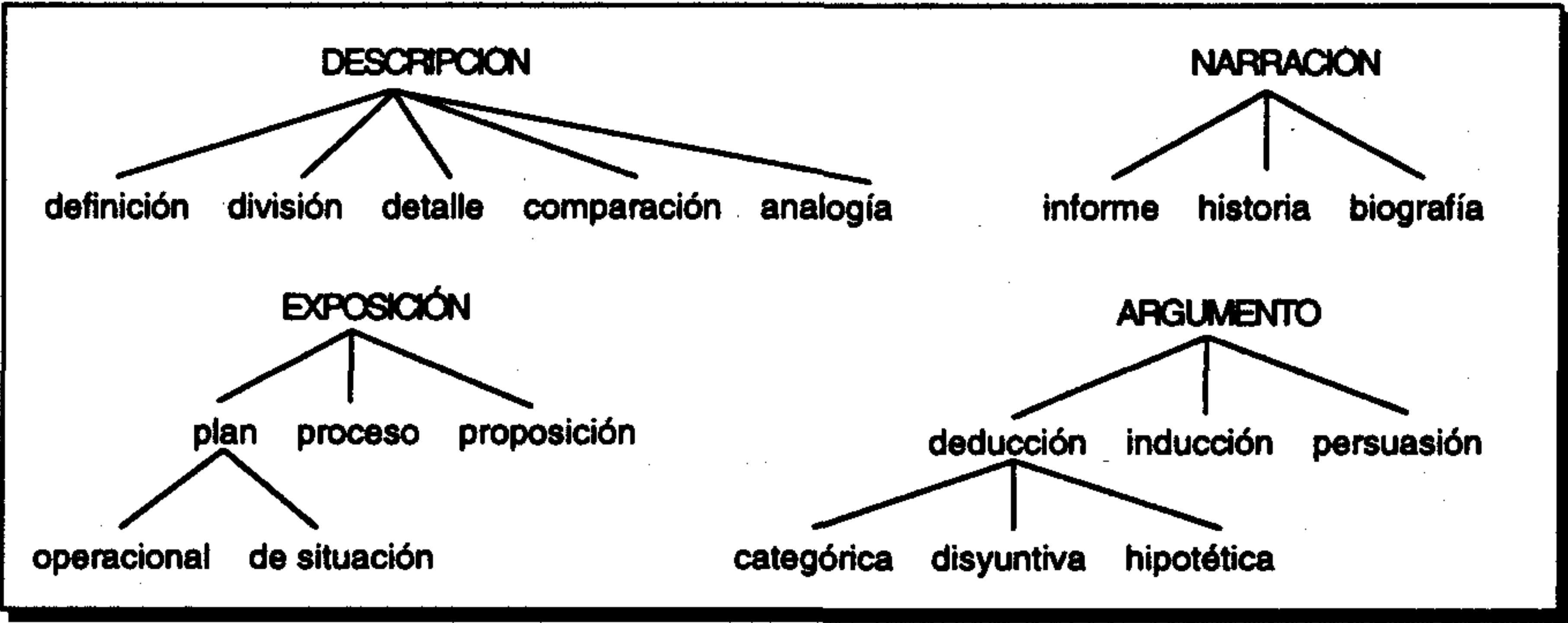


Figura 3.5: Clasificación de los tipos de explicación

A partir de esta clasificación, Maybury propuso una teoría de actos comunicativos que caracteriza un texto en términos de tres clases de actos: (i) actos retóricos, (ii) actos no-locutivos, y (iii) actos locutivos. Los actos retóricos definen actos comunicativos de alto nivel que se corresponden con las cuatro categorías de explicación: describir, narrar, exponer y argumentar. A un nivel inferior de abstracción están los actos no-locutivos, como informar, requerir, solicitar y avisar; y finalmente, los actos locutivos o actos de habla superficiales que son aquellos que se emplean para llevar a cabo actos no-locutivos, como afirmar, pedir, ordenar, recomendar. La figura 3.6 muestra un ejemplo de esta clasificación para el acto retórico Describir.



Aunque, al igual que las propuestas presentadas anteriormente, su visión de la planificación de presentaciones está basada en el análisis de la estructura de los textos, presenta diferencias respecto a estos otros modelos. Una de ellas es que Maybury utiliza los predicados retóricos como los elementos constitutivos de un texto pero no los estructura como esquemas retóricos [McKeown et al., 90] sino que los organiza jerárquicamente para definir un plan de actos comunicativos. Por otra parte, en el momento de diseñar la estructura de una explicación, [Hovy, 91] por ejemplo, utiliza predicados retóricos para construir una estructura retórica sobre un conjunto dado de proposiciones; [Moore, Paris, 93] también utiliza este modelo pero en su caso las hojas del árbol que define la estructura de la presentación son actos no-locutivos que tienen también proposiciones asociadas (ver figura 3.7 (a)).

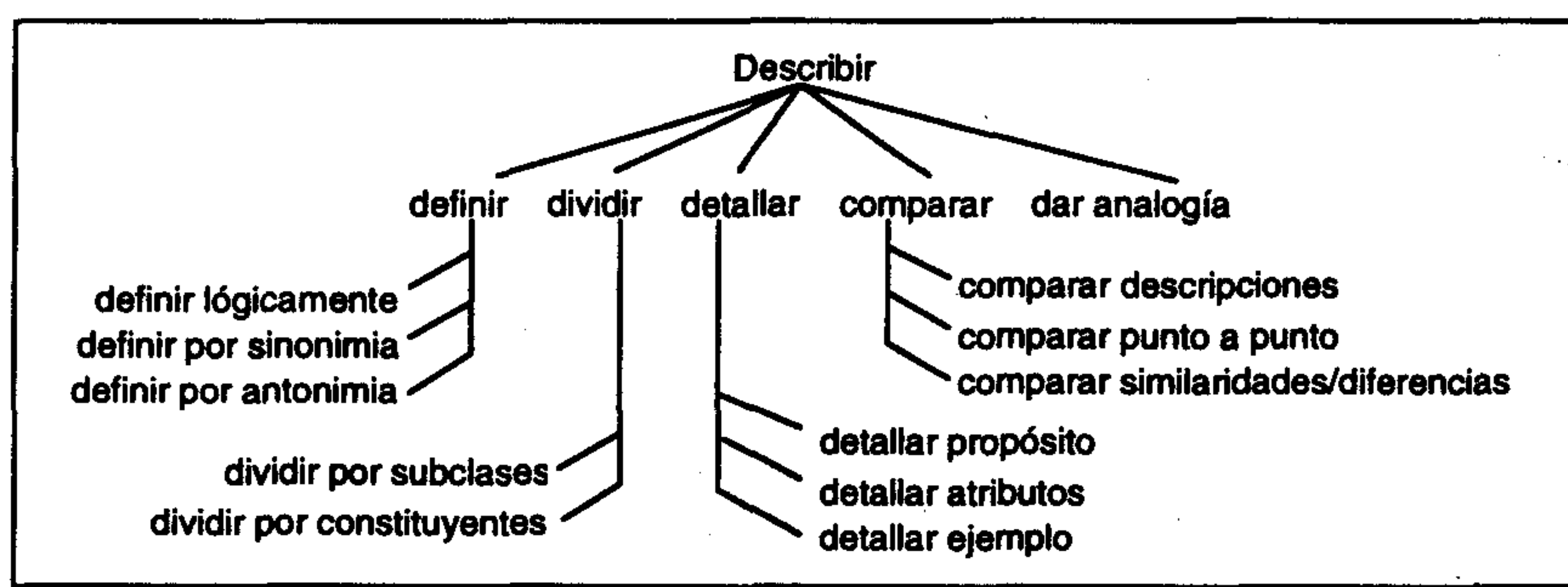


Figura 3.6: Clasificación de actos comunicativos para la descripción

Los modelos de Hovy y Moore se fundamentan en las relaciones entre las partes de un texto, pero no consideran aspectos asociados a los predicados retóricos como la estructura jerárquica, las precedencias o las diferencias epistemológicas. Estos aspectos son los que contempla Maybury en su representación de los predicados retóricos, en la que cada acto comunicativo de la jerarquía tiene asociada información explícita sobre su efecto, relaciones jerárquicas y orden de los subactos (ver figura 3.7 (b)). Al disponer de información explícita sobre el efecto esperado de cada acto comunicativo, la estructura de una presentación puede ofrecer dos visiones: una que muestra la descomposición de las acciones comunicativas y otra que muestra la descomposición de los efectos de esas acciones. Con esta segunda visión se dispone de información sobre el efecto que cada una de las partes o sentencias del texto provoca en el estado cognitivo del usuario, es decir, en el modelo del usuario.

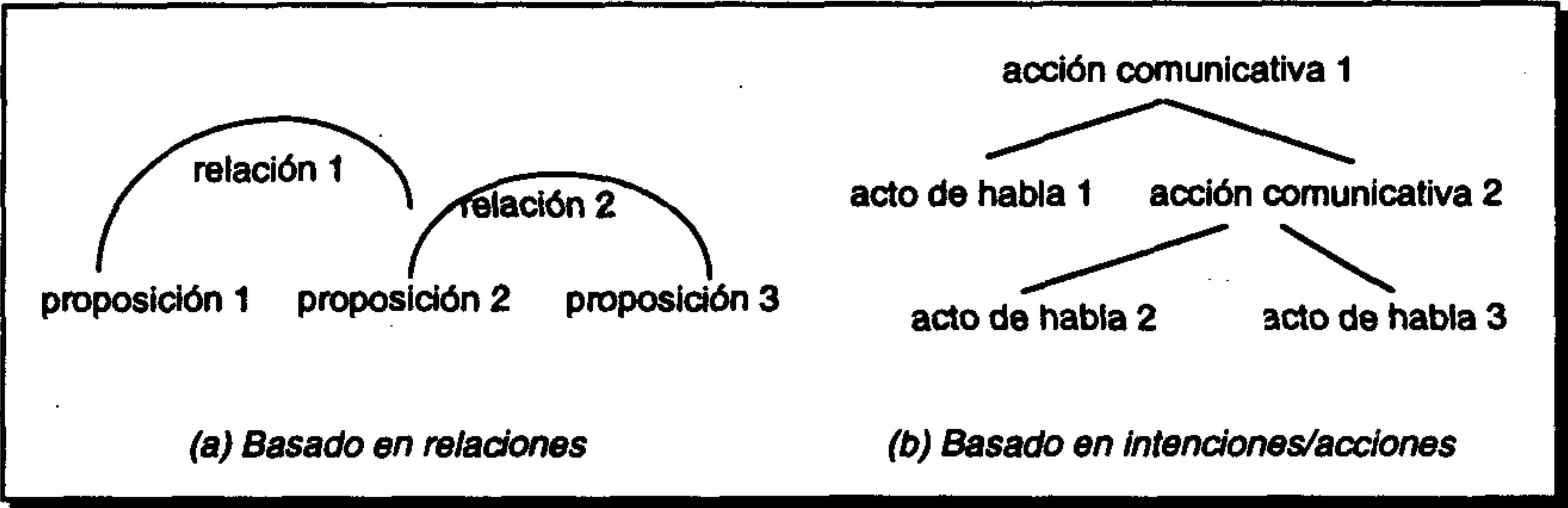


Figura 3.7: Contenido y estructura de las presentaciones textuales

Esta teoría de la organización de presentaciones en lenguaje natural se ilustró con el sistema planificador de texto TEXPLAN (*Textual EXplanation PLANer*). En TEXPLAN, cada acto comunicativo se representa como un operador de planificación que incluye información sobre (ver figura 3.8):

- precondiciones que deben verificarse para que pueda ser utilizado,
- restricciones sobre aspectos físicos o cognitivos de los modelos del dominio o del usuario,
- efecto esperado en el modelo de usuario, y
- descomposición en subacciones.

El hecho de representar explícitamente el efecto provocado por un operador es también un aspecto diferenciador del modelo de Maybury, ya que, por ejemplo, [McKeown et al., 90] no lo representa, y [Moore, Paris, 93] sí lo hace pero mezclando el estado de conocimiento del usuario con objetivos retóricos asociados al operador.

NAME	describe_by_defining
HEADER	Describe( <i>speaker</i> , <i>hearer</i> , <i>entity</i> )
CONSTRAINTS	Entity?( <i>entity</i> )
PRECONDITIONS	
ESSENTIAL	Know-About( <i>speaker</i> , <i>entity</i> ) ^ Want( <i>speaker</i> , Know-About( <i>hearer</i> , <i>entity</i> ) )
DESIRABLE	Not Know-About( <i>hearer</i> , <i>entity</i> )
EFFECTS	Know-About( <i>hearer</i> , <i>entity</i> )
DECOMPOSITION	Define( <i>speaker</i> , <i>hearer</i> , <i>entity</i> )

Figura 3.8: Ejemplo de operador de planificación de presentaciones (adaptado de [Maybury, 92])

El funcionamiento del sistema TEXPLAN está dividido en dos procesos básicos: planificación de texto y realización lingüística (ver figura 3.9). La



planificación de texto se inicia cuando la aplicación o el usuario plantean un objetivo de discurso al gestor del diálogo, expresado en términos del efecto que se espera produzca sobre el conocimiento del usuario, sus objetivos, creencias o planes. Utilizando un algoritmo de unificación, busca en la biblioteca de operadores todos aquellos cuya cabecera equipara con el objetivo. Los operadores seleccionados se instancian ligando las variables con los valores que toman en la especificación del objetivo, y los que satisfagan las restricciones y precondiciones esenciales, se ordenan atendiendo a ciertas métricas de preferencia (texto y gráficos es mejor que sólo texto, se prefieren aquellos con menos subobjetivos, que verifican todas las precondiciones, etc). Trabajando con esta lista ordenada de operadores, el planificador se asegura de que las precondiciones se cumplan y ejecuta la descomposición en subobjetivos hasta que se cumple el objetivo inicial. Esta ejecución puede suponer procesar operadores especiales (por ejemplo, que permiten opcionalidad), cuantificadores, así como distinguir entre subobjetivos y actos primitivos. El resultado es una descomposición jerárquica de planes que se linealiza con un recorrido en profundidad, que es el que finalmente determina el orden de ejecución de los actos primitivos lingüísticos presentes en las hojas del árbol de planes. Por último, el realizador lingüístico traduce el plan anterior a texto.

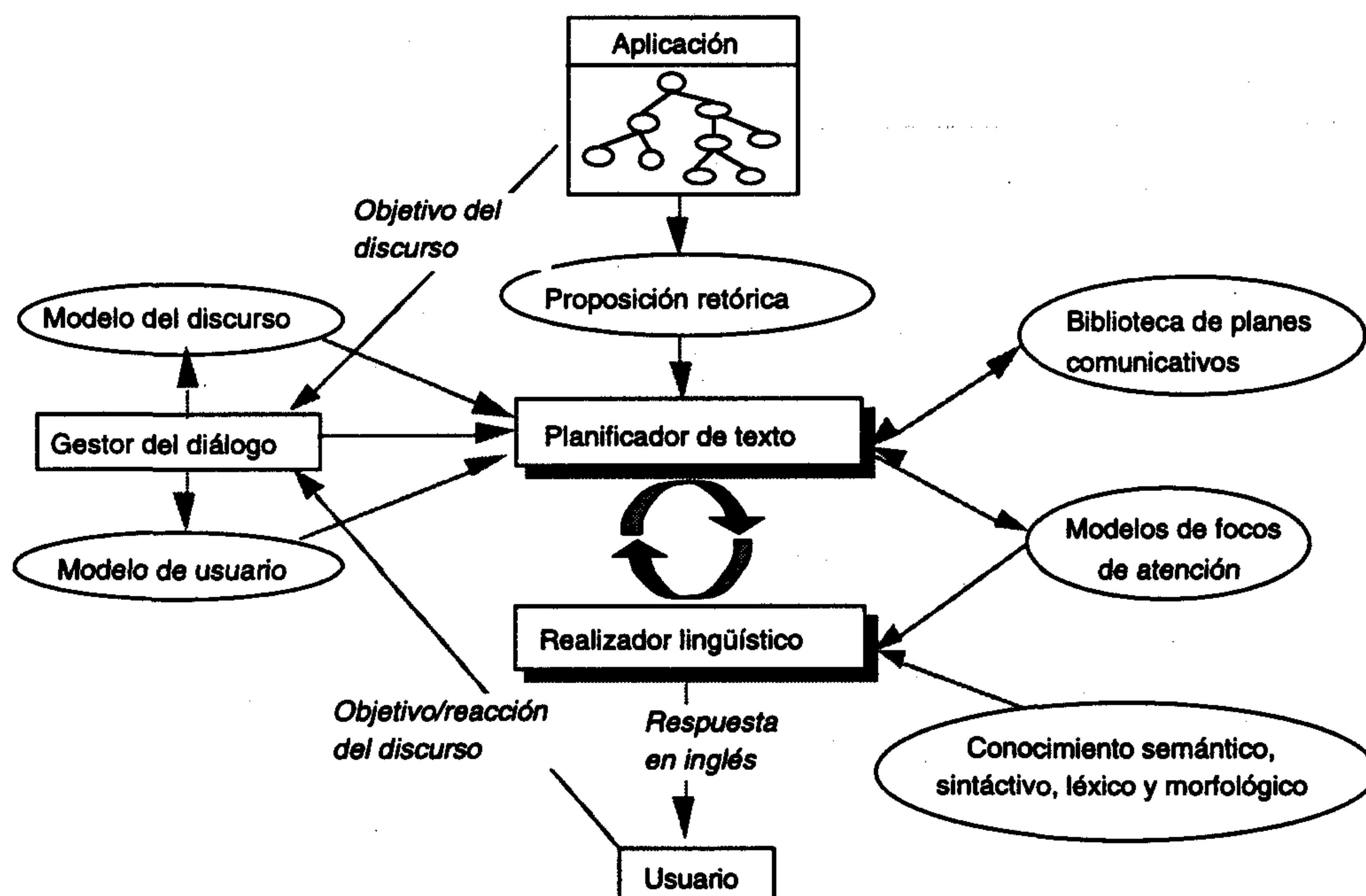


Figura 3.9: Arquitectura del sistema TEXPLAN  
(adaptado de [Maybury, 92])

Cuando el usuario lee el texto generado por el realizador lingüístico, tiene la oportunidad de aceptarlo o indicar alguna reacción en un número de formas predeterminadas. Basándose en esa reacción, el gestor del diálogo puede actualizar el modelo de usuario de acuerdo con los efectos esperados del texto, rediseñar el plan para obtener otra explicación que responda al mismo objetivo de discurso, plantear un nuevo objetivo de discurso al planificador de texto, o acabar la interacción.

### **3.3 Presentaciones Inteligentes Multimedia**

Tras el avance experimentado por los interfaces de usuario con lenguaje natural, descrito en las secciones anteriores, en los últimos años la introducción de diferentes dispositivos multimedia en el desarrollo de los interfaces proporcionó nuevos elementos de mejora del nivel de interacción entre usuarios y sistemas. Esta mejora se produce porque la diversificación de los canales de comunicación permite aprovechar las capacidades expresivas de cada uno de los medios, haciendo que las carencias de unos las suplan las potencialidades de los otros. Aunque de esta manera, se hace posible asociar a cada parte de la información a transmitir el medio de comunicación más adecuado en función de la naturaleza de esa parte, se aumenta la complejidad del diseño de los interfaces puesto que han de coordinarse adecuadamente los distintos medios empleados en una presentación. La complejidad de estos interfaces ha hecho que actualmente se esté trabajando en el establecimiento de una arquitectura multimedia de referencia común a todos los procesos implicados en la generación de presentaciones multimedia [Bordegoni et al., 98]. Esta sección muestra las soluciones más significativas desarrolladas en esta línea.

#### **3.3.1 Las acciones retóricas ampliadas de Maybury**

[Maybury, 93] presenta una ampliación del modelo teórico instrumentado en TEXPLAN para generar explicaciones multimedia y no sólo textuales. Para ello, extiende el conjunto de actos comunicativos considerados en TEXPLAN con otros específicamente dedicados a representar acciones gráficas o auditivas; lo que produce una nueva clasificación de los actos comunicativos: actos retóricos independientes del medio utilizado (por ejemplo, identificar,



describir), actos lingüísticos (locutivos y no-locutivos), actos gráficos (por ejemplo, actos deícticos), y otros como actos auditivos no lingüísticos y actos físicos (por ejemplo, gestos).

El conocimiento de presentación y el funcionamiento del planificador son prácticamente los mismos del sistema TEXPLAN inicial con la salvedad del conocimiento adicional añadido para generar las presentaciones multimedia. Este conocimiento adicional se introduce a dos niveles:

- (i) en la formulación de los operadores, que ahora pueden incluir aspectos gráficos tanto en las restricciones como en las precondiciones y descomposición en suboperadores (ver figura 3.10); y
- (ii) en la definición de las métricas de preferencia utilizadas para seleccionar operadores.

El enfoque planteado por Maybury para generar las presentaciones multimedia plantea una aproximación de “grano gordo” en el diseño de la presentación que aunque evita algunos de los problemas de coordinación asociados al uso de varios dispositivos de comunicación resta flexibilidad en la generación de las explicaciones. Por ejemplo, la coordinación entre la selección del contenido de la presentación y el modo de presentarla está contemplada explícitamente en la definición de los operadores y las métricas de preferencia, ya que éstos contienen conocimiento no sólo sobre qué información presentar sino también sobre combinaciones adecuadas de los modos de presentarla.

NAME	Identify-location-linguistically-&-visually
HEADER	Identify( <i>S</i> , <i>H</i> , <i>entity</i> )
CONSTRAINTS	Cartographic-Entity?( <i>entity</i> )
PRECONDITIONS	Visible( <i>entity</i> ) $\wedge$ Wants( <i>S</i> , Know( <i>H</i> , Location( <i>entity</i> )))
EFFECTS	Know( <i>H</i> , Location( <i>entity</i> ))
DECOMPOSITION	Indicate-Deictically( <i>S</i> , <i>H</i> , <i>entity</i> ) Assert( <i>S</i> , <i>H</i> , Location( <i>entity</i> ))

Figura 3.10: Operador para presentaciones gráficas/textuales  
(de [Maybury, 93])

Por otra parte, la realización práctica de este modelo se ha restringido a su aplicación en un sistema de orientación cartográfica en el que se pueden



generar presentaciones animadas de rutas entre dos puntos de un mapa, intercalando frases con parpadeos y coloreo de tramos. No hay un modelo de resolución de problemas bajo este interfaz, la información que se presenta en pantalla se obtiene invocando funciones incluidas en la definición de los operadores que devuelven el camino más corto entre dos puntos.

En la actualidad el grupo de Maybury trabaja en la indexación inteligente de múltiples medios, considerando la explotación de otros medios como fuentes de información; por ejemplo, procesando información expresada en lenguaje hablado o escrito asociada a imágenes en movimiento de manera que pueda habilitarse un acceso dirigido por esa información a las imágenes [Maybury et al., 97].

### 3.3.2 WIP

WIP [André et al., 93], [Wahlster et al., 93] es un prototipo de un sistema generador de presentaciones multimedia basado en el conocimiento que adapta sus presentaciones a las características de la audiencia. Como en el caso de Maybury, su desarrollo como un planificador se fundamenta en el principio de que tanto la generación de texto y articulación de diálogos, como el diseño de gráficos y documentos multimodales puede considerarse una secuencia de actos que intentan alcanzar unos objetivos comunicativos concretos.

El modelo teórico elegido para caracterizar la estructura de un documento ilustrado es la Teoría de Estructuras Retóricas (RST, [Mann, Thompson, 87]). Con esta teoría, como se vió en el apartado 3.1.2, la estructura de un documento se define en base a una estructura jerárquica de actos comunicativos más una representación de las relaciones de unos actos con otros. Las relaciones entre actos comunicativos definen los niveles superiores de la jerarquía e incluyen, por ejemplo: *compara* (dos partes del documento ofrecen una comparación de varios conceptos), *elabora* (una parte del documento proporciona detalles sobre otra parte), *capacita* (el texto o el gráfico proporcionan información adicional para facilitar al operador la realización de una acción determinada), etc.

Respecto a la asignación del modo de presentación al contenido a presentar, ésta se realiza teniendo en cuenta la naturaleza de la información. Por ejemplo, para información concreta (propiedades visuales de conceptos) o espacial,



normalmente el modo de presentación elegido es el gráfico; mientras que para información temporal o cuantificación la descripción textual se considera más adecuada. El diseño final de la estructura del documento a presentar es el resultado de un proceso de planificación, con operadores que incorporan los criterios anteriores y que dan lugar a un plan en el que se reflejan los contenidos proposicionales de las partes del documento, el efecto buscado con cada una de las partes, así como relaciones retóricas entre ellas.

Los operadores de planificación empleados definen estrategias de presentación, en la misma línea del trabajo de Moore y Paris (apartado 3.2.1), con los siguientes elementos:

- *nombre y cabecera*, que representa un objetivo comunicativo de alto nivel;
- *efecto*, que define un cambio sobre el estado de conocimiento del usuario;
- *condiciones de aplicabilidad*, que describen cuándo el operador puede utilizarse y restringen los valores de las variables a instanciar;
- especificación de *actos principales y secundarios*, que definen la descomposición en subactos necesaria para llevar a cabo el objetivo expresado en la cabecera.

La forma en la que se definen estos operadores deja abierta la posibilidad de definir las estrategias de presentación a diferentes niveles de especificidad anticipando o no importantes decisiones de diseño, como por ejemplo la selección del modo de presentar la información. Si se restringen las decisiones de diseño, podrían evitarse situaciones en las que una decisión ha de retraerse porque no puede llevarse a cabo, pero por contra, podrían introducirse restricciones innecesarias a los posibles diseños. Cuando se dispone de más de una estrategia u operador para alcanzar un objetivo, se utilizan metareglas para hacer la selección. La ejecución del plan la llevan a cabo, en paralelo, módulos independientes de generación de presentaciones: un generador de textos y un generador de gráficos.

La figura 3.11 muestra un ejemplo de una estrategia de presentación orientada a conseguir que el usuario lleve a cabo una acción, que se expresa en términos similares a los utilizados por Hovy en sus operadores (ver apartado 3.1.2). En este ejemplo puede observarse cómo los actos principales se realizan usando texto, mientras que el modo a utilizar con los actos secundarios está abierto. Esta estrategia considera tres clases de acciones: la acción elemental *S(urface)-*



*Request*, tres acciones asociadas con las que especificar la acción anterior y su semántica (*Activate*) y, dos acciones comunicativas complejas (*Motivate* y *Enable*).

NAME	Request-Enable-Motivate
HEADER	(Request P A ?action Text)
EFFECT	(BMB P A (Goal P (Done A ?action)))
APPLICABILITY CONDITIONS	(And (Goal P (Done A ?action)) (Bel P (Complex-Operating-Action ?action)) (Bel P (Agent ?agent ?action)) (Bel P (Object ?object ?action)))
MAIN ACTS	(S-Request P A (?action-spec (Agent ?agent-spec) (Object ?object-spec))) (Activate P A (Action ?action) ?action-spec Text) (Activate P A (Agent ?agent) ?agent-spec Text) (Activate P A (Object ?object) ?object-spec Text)
SUBSIDIARY ACTS	(Motivate P A ?action ?mode-1) (Enable P A ?action ?mode-2)

Figura 3.11: Ejemplo de estrategia de presentación

El proceso de planificación de la presentación sigue el mismo esquema básico del modelo de Maybury: para cada objetivo de presentación, se buscan operadores cuya cabecera o cuyo efecto equiparen con el objetivo. De éstos, se seleccionan aquellos para los que se satisfacen las condiciones de aplicabilidad. Si son más de uno, entonces se aplican metareglas que los ordenan por prioridades. Una vez seleccionado uno de los operadores, sus actos principales y secundarios se establecen como nuevos subobjetivos o si se trata de actos comunicativos elementales, se añaden a las colas de tareas de los generadores específicos del modo de presentacion correspondiente. Si un subobjetivo no puede alcanzarse, se da marcha atrás y se busca una estrategia alternativa. Este proceso termina si todos los objetivos se han expandido hasta el nivel de actos elementales que pueden realizarse con los generadores de texto o gráficos. El resultado del proceso de planificación es un plan detallado sobre el estilo de presentar cada parte de la información a transmitir representado como un grafo dirigido acíclico.

Sin embargo, el proceso anterior se complica ya que para asegurar que los fragmentos del documento están adecuadamente enlazados deben considerarse las dependencias entre la planificación del contenido, la selección del modo de presentarlo, y la generación de la presentación. Los dos primeros están ligados por el propio mecanismo de planificación, ya que los operadores



y las metareglas usadas por el planificador contienen no sólo conocimiento sobre qué presentar sino también conocimiento sobre combinaciones adecuadas de modos de presentación. Por contra, la planificación y la realización de la presentación se llevan a cabo con componentes separados que acceden a fuentes de conocimiento distintas. Esta modularización hace posible el procesamiento en paralelo de la selección y presentación de la información, por lo que cuando el planificador ha decidido qué generador debe encargarse de presentar una pieza de información, ésta es enviada al mismo, a la vez que va incorporando los resultados de los generadores sobre otras partes de la presentación. La arquitectura general del planificador de presentaciones de WIP se muestra en la figura 3.12.

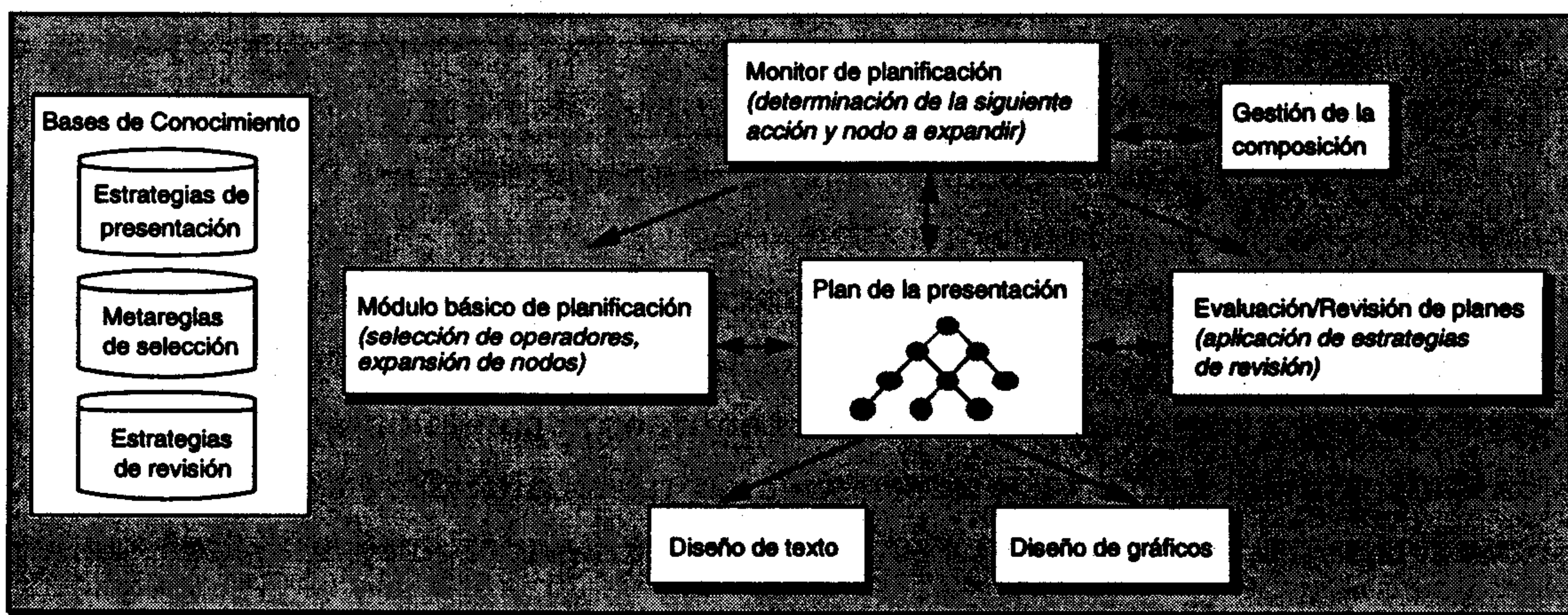


Figura 3.12: Arquitectura del planificador de presentaciones

El flujo de información entre estos tres componentes permite que el planificador propague cambios introducidos en el plan inicial, tras recibir una respuesta de un generador, para mantener la consistencia del documento. Por ejemplo, si el generador de texto ofrece una expresión que referencia una parte de un objeto mostrado en pantalla y este objeto ya no está o ha sido modificado por el generador de gráficos entonces el planificador tiene que modificar el plan para construir una nueva descripción del objeto y hacer que el generador de texto reconsidere su expresión. Además, al incorporar en la definición de los operadores dependencias entre estrategias de presentación también se hace necesario propagar las decisiones que se vayan haciendo sobre el modo de presentar partes del documento. Estas actividades se llevan a cabo mediante un módulo responsable de la evaluación y revisión de los planes, el cual es



gestionado desde un módulo de monitorización que es el que hace posible alternar los procesos de expansión y revisión de planes.

En resumen, WIP propone un modelo de planificación para generar automáticamente documentos ilustrados que soporta la transferencia de información entre el planificador del contenido y los generadores especializados en distintos modos de presentación, y que permite la evaluación y revisión del plan global en cada paso de planificación. En los últimos tiempos, el grupo de WIP ha estado trabajando en la extensión del sistema WIP al sistema PPP (*Personalized Plan-Based Presenter*) [Rist et al., 97] que permite la interacción del usuario durante la presentación multimedia. El planificador de presentaciones de WIP se extendió para generar planes de presentación condicionados al comportamiento del usuario en el diálogo. Así, el usuario puede cambiar parámetros de la presentación, encadenar preguntas y controlar la salida gráfica mediante un sistema de menús. En PPP, las estrategias de presentación de WIP se aumentaron con restricciones temporales cuantitativas y cualitativas de manera que se hace posible la sincronización automática de todos los actos de la presentación. Como el tiempo consumido por las interacciones del usuario no puede ser previsto con seguridad, PPP puede revisar y adaptar su plan preliminar de presentación si cree percibir (mediante un componente de modelización de usuarios llamado PEDRO) que la presentación planificada podría no ser comprendida adecuadamente por el usuario.

A diferencia de WIP, el sistema PPP no sólo construye documentos multimedia sino que planifica cómo este material debe ser presentado para un usuario concreto. PPP personaliza una presentación desde dos puntos de vista: (i) por una parte, adaptando su contenido, expresión y visualización a las características de un usuario, y por otra, (ii) personificando el sistema de presentación y el modelo de usuario en un agente interfaz llamado *Persona* que aparece como un dibujo animado que comenta y explica las presentaciones generadas.

### 3.3.3 COMET

COMET (*COordinated Multimedia Explanation Testbed*) [Feiner, McKeown, 93] es un sistema basado en el conocimiento que produce presentaciones interactivas coordinadas que combinan texto y gráficos tridimensionales. El dominio de



desarrollo y uso de COMET ha sido el mantenimiento y reparación de radios militares emisoras/receptoras. El usuario interactúa con el sistema mediante un menú sencillo en el que, tras seleccionar un conjunto de síntomas, puede solicitar que el sistema le dé instrucciones para llevar a cabo pruebas de diagnóstico o bien que el sistema realice el diagnóstico e identifique el problema y sus causas.

La arquitectura de COMET consiste en un conjunto de procesos paralelos que cooperan para diseñar una presentación (ver figura 3.13). A partir de una solicitud del usuario, el planificador del contenido utiliza planes de texto o esquemas, en la línea de [McKeown, 85], para determinar qué información debería ser incluida en la respuesta. Para ello COMET utiliza tres fuentes de conocimiento del dominio distintas:

- (i) una representación estática de los objetos y las acciones del dominio,
- (ii) una base de reglas de diagnóstico, y
- (iii) una base de conocimiento geométrico detallada necesaria para generar los gráficos.

También mantiene un modelo de usuario y un modelo del diálogo mantenido en la sesión. Estas fuentes de conocimiento son utilizadas por todos los componentes del sistema para construir la respuesta, y no sólo por el planificador de contenido.

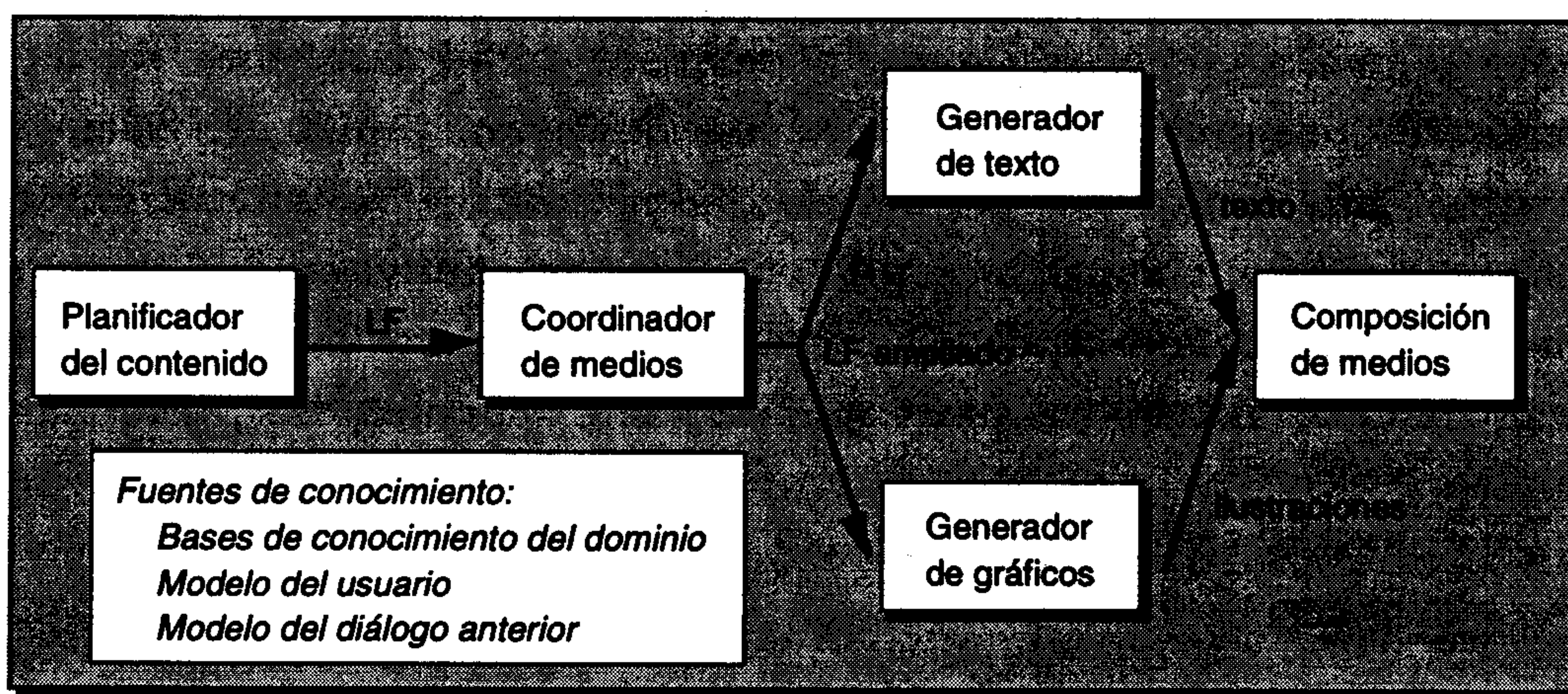


Figura 3.13: Arquitectura del sistema COMET



El planificador de contenido define el contenido de la explicación representado como una jerarquía de formularios lógicos o LFs (*logical forms*) [Allen, 87]. Un LF consiste en una colección de pares atributo-valor que describen acciones o conceptos del dominio. Los LFs se le pasan al coordinador de medios que los refina ampliando su descripción con anotaciones sobre los modos que han de emplearse para presentar cada una de las partes de la información a transmitir. El coordinador de medios realiza un análisis detallado del LF que recibe para clasificar en su contenido hasta seis tipos distintos de información. Cada uno de estos tipos tiene asociado el modo de presentación adecuado. Así, atributos físicos o de localización son presentados exclusivamente con gráficos, las acciones abstractas y las relaciones entre acciones (como causalidad) se expresan sólo con texto, y las acciones simples o compuestas se presentan con ambos medios. Lo que hace el coordinador es una asociación entre los tipos de información y los medios que han de presentarla. La figura 3.14 muestra un ejemplo de un LF ampliado.

```
((cat if)
 (directive-act substeps)
 (substeps
  (((process-type action)
   (process-concept c-push)
   (mood non-finite)
   (speech-act directive)
   (function ((type substeps)
    ((media-text yes)
     (media-graphics no))))
   (roles
    ((medium
     ((object-concept c-button-or)
      (roles
       ((location ((object-concept c-location)
        ((media-graphics yes)
         (media-text no)))
        (size ((object-concept c-size)
         (media-graphics yes)
          (media-text no))))))))
    ))
   ))
  ))
))
```

Figura 3.14: Ejemplo de LF ampliado  
(de [Feiner, McKeown, 93])

Estos LFs ampliados los reciben tanto el generador de texto como el de gráficos, que tras generar los fragmentos de texto y gráficos necesarios, los ligam a los puntos correspondientes del formulario. El generador de texto elabora presentaciones de los segmentos del formulario a los que el



coordinador les ha asignado como medio el texto. Primero determina el número de frases necesarias para presentar ese segmento de información y su tipo (simple, compuesto, declarativo o imperativo), y después selecciona verbos para expresar las acciones del formulario y sustantivos o modificadores para los objetos. Esta selección la realiza teniendo en cuenta restricciones de la base de conocimiento, del modelo de usuario o del discurso previo. Finalmente, construye la estructura sintáctica de cada frase y linealiza el árbol resultante como una frase compuesta. Por su parte, el generador de gráficos busca satisfacer los objetivos comunicativos especificados en el LF que recibe como entrada, empleando una base de reglas en la que cada objetivo comunicativo que puede aparecer en un formulario tiene asociado al menos una regla. Cada regla de diseño invoca un conjunto de estrategias de estilo que especifican efectos visuales de alto nivel, que a su vez se llevan a cabo mediante otras reglas de más bajo nivel. El conocimiento utilizado por este componente incluye criterios de evaluación del éxito de la presentación o su visualización; por ejemplo, ha de asegurarse que un objeto no esté oculto por otro, o hacer que destaque frente al resto aumentando su luminosidad. Las salidas de ambos generadores las procesa un componente de composición de medios que da forma a la presentación final.

Una singularidad de COMET es que sus LFs dan un servicio similar al de las pizarras, ya que cada componente puede leer y modificar los LFs, enriqueciéndolos y refinándolos hasta que la explicación está completa. Este uso de los formularios hace que sirva de vehículo de comunicación entre los generadores de texto y gráficos. Por ejemplo, cuando se examinan cuáles son las posibilidades expresivas más adecuadas a cierta pieza de información, un generador puede observar las decisiones tomadas para otros medios leyendo sus formularios lógicos ampliados y utilizar esta información para tomar sus propias decisiones.

El modelo de coordinación de medios que utiliza COMET es bastante diferente del de WIP, aun cuando ambos generan presentaciones parecidas para el mismo tipo de aplicaciones, en las que se describen objetos y acciones en tres dimensiones. WIP le pasa a sus generadores las piezas de información a presentar de forma secuencial mientras que COMET les ofrece toda una estructura de información que incluye todas las piezas que ha de procesar, de una vez, más las que ha de procesar el otro generador. La asignación del modo de presentación a una pieza de información en WIP se hace dinámicamente

evaluando el éxito de las presentaciones ya generadas anteriormente. En COMET esta asignación se hace simultáneamente sobre todas las piezas de información. Así, en ambos casos un generador sabe lo que ha hecho el otro, pero COMET proporciona a sus generadores más información del contexto global de la explicación que pueden tener en cuenta para tomar sus decisiones iniciales. Las ventajas de disponer de un repositorio común de información sobre la presentación, en el que los generadores pueden hacer anotaciones sobre la realización de sus procesos hace posible que:

- texto y gráficos se influyeran mutuamente permitiendo referencias cruzadas del texto a los gráficos,
- los objetivos comunicativos estén separados de los recursos empleados para llevarlos a cabo, y
- se disponga de un mecanismo para comunicar los generadores de texto y gráficos.

La línea de investigación actual se dirige hacia la generación coordinada de medios temporales, como lenguaje hablado y gráficos animados mediante una negociación en dos fases controlada desde el coordinador de medios, que está siendo experimentada en el sistema MAGIC (*Multimedia Abstract Generation for Intensive Care*) [Dalal et al., 96], que genera informes multimedia para cuidadores de hospitales. En la primera fase, los generadores de lenguaje y gráficos le pasan al coordinador de medios una lista ordenada de las secuencias de ordenación preferidas, el cual utiliza un resolutor de restricciones temporales para encontrar un orden compatible para ambos tipos de medios. En la segunda, los generadores de lenguaje hablado y gráficos animados plantean restricciones sobre la duración de sus presentaciones de acuerdo con el orden establecido, que nuevamente son filtradas por el coordinador de medios.

### 3.4 Modelos de Usuario

La necesidad de definir modelos de usuario surge, como se ha visto anteriormente, de la evolución de los sistemas con interfaces en lenguaje natural. Durante los años 70 se desarrollaron muchos interfaces de lenguaje natural de propósito específico que respondían mecánicamente a las preguntas



del usuario, el cual tenía que encontrar una estrategia en la formulación de sus preguntas adecuada para que el sistema le proporcionase la información que resolvería su problema, sin esperar ninguna ayuda por parte del sistema.

En los años 80 apareció una nueva clase de sistemas dialogantes orientados a tareas caracterizados por roles predefinidos que el usuario y el sistema debían asumir durante el desarrollo de un diálogo. El usuario de estos sistemas no necesitaba preparar una estrategia de preguntas porque el sistema trataba de reconocer la intención del usuario con respecto al dominio del discurso, de manera que se estableciese un diálogo más cooperativo. Es en escenarios de interacción como los que presentan estos sistemas donde surgió la necesidad de construir modelos explícitos de las creencias, los objetivos y los planes del usuario, para poder ofrecer un amplio espectro de comportamiento dialogante cooperativo. De esta forma, ya no es sólo una tarea del usuario el construir un modelo mental de las funcionalidades del sistema, sino que el sistema debe formularse hipótesis sobre lo que el usuario cree, quiere y planea, es decir construir un modelo del usuario. En este sentido, [Verdejo, 92] ofrece una visión general de los elementos más significativos a considerar en la caracterización de un modelo de usuario.

Entonces, puede definirse un modelo de usuario como *una fuente de conocimiento, en un sistema con capacidades de diálogo, que contiene hipótesis explícitas de todos los aspectos del usuario que podrían ser relevantes para el comportamiento dialogante del sistema* [Wahlster, Kobsa, 86]. Estos aspectos pueden referirse al conocimiento del usuario sobre el dominio de la aplicación, a rasgos de su personalidad [Rich, 89], a sus preferencias [Morik, 89], [Morik, Rollinger, 85], o a sus planes y objetivos [Carberry, 89], [Johnson, Soloway, 84], [Litman, Allen, 84], [Chin, 89].

Según [Kass, Finin, 86], los modelos de usuarios sólo resultan útiles a sistemas con alguna de las siguientes características:

- sistemas que intentan adaptar su comportamiento a los usuarios individuales,
- sistemas que asumen la responsabilidad (o comparten esa responsabilidad con el usuario) de asegurar el éxito de la comunicación usuario-sistema,

- sistemas cuyos posibles usos o cuyo conjunto de usuarios potenciales es variado.

Los aspectos fundamentales en la construcción de modelos de usuario son:

- (i) qué información ha de incluirse en un modelo de usuario, y
- (ii) cómo ha de configurarse y mantenerse ese modelo.

Sin embargo, estas actividades plantean numerosos problemas cuya solución ha de establecerse ad hoc dependiendo del tipo de sistema en el que el modelo de usuario se integre. Por otra parte, dado el coste de construcción y mantenimiento de los modelos de usuarios, éstos sólo se justifican si los sistemas para los que van destinados tienen un abanico de usuarios potenciales variado o un número de aplicaciones diversas.

### 3.4.1 Problemática de la modelización de usuarios

La construcción y mantenimiento de un modelo de lo que un usuario conoce presenta varias dificultades que se describen a continuación:

- *Recopilación de la información.* El sistema debe primero determinar qué es lo que sabe el usuario y después representar esta información internamente. La identificación de lo que sabe un usuario sobre un dominio podría hacerse con una consulta exhaustiva, pero en la mayoría de los dominios este planteamiento es muy costoso en tiempo y los usuarios no están dispuestos a llevarlo a cabo. Sin embargo, el proceso de adquisición de conocimiento en un dominio suele hacerse con un orden predecible, por lo que, en principio, un sistema podría hacer inferencias sobre lo que un usuario sabe o no sabe basándose en un modelo parcial del mismo, aunque esto no deja de ser un aspecto problemático.
- *Manejo de la incertidumbre.* Hacer inferencias sobre un usuario basándose en un limitado conocimiento inicial sobre el mismo es una forma de razonamiento por defecto. Como este razonamiento no es siempre válido, el sistema ha de tener capacidad para juzgar la certeza de sus deducciones. Además, cuando el sistema adquiere mayor información sobre el usuario el modelo debe actualizarse, y podría ser que la nueva información contradijese afirmaciones hechas previamente, por lo que el mantenimiento de un modelo de usuario no es simple.



- *Organización del conocimiento.* En ocasiones, pueden identificarse grupos de deducciones o inferencias asociados a diferentes etapas del aprendizaje del usuario en el dominio de la aplicación. Usuarios en una misma etapa de aprendizaje suelen tener un conocimiento similar [Kay, Black, 85], por lo que, cuando un sistema ha determinado el nivel de un usuario puede asumir que éste conoce lo mismo que otros usuarios a ese mismo nivel. Entonces, sería deseable que estos sistemas agrupasen este tipo de deducciones de forma sencilla y eficiente.

### 3.4.2 Representación de modelos de usuario

Una vez que el conocimiento de un usuario ha sido determinado queda el problema de representar internamente ese conocimiento. Los modelos de usuario se expresan mediante representaciones formales que se eligen en función del propósito del sistema y la demanda de poder expresivo requerida para el modelo. A continuación se describen algunas de las soluciones de representación de modelos de usuario:

#### Modelos *overlay*

La forma más simple de representar la información recopilada sobre un usuario es utilizando un modelo *overlay* [Carr, Goldstein, 77], en el que el conocimiento de un usuario se representa como un subconjunto del conocimiento del sistema. Una variante de este modelo es representar al usuario como una colección de diferencias con respecto al conocimiento de un experto que se ve frecuentemente aumentada con una librería de procedimientos o reglas comunes entre los usuarios ([Brown, Burton, 78], [Stevens et al., 79], [Sleeman, Smith, 81], [Johnson, Soloway, 84], [Reiser et al., 85], [Anderson et al., 85]).

Los modelos *overlay* no tienen capacidad de predicción ya que no representan el orden en el que los usuarios van adquiriendo nuevo conocimiento sobre el dominio. Estos, o variantes de los mismos, han sido utilizados normalmente en sistemas tutoriales inteligentes ([Brown, Burton, 76], [Sleeman, Brown, 82], [Kass, 89]) en los que debe mantenerse un modelo de usuario preciso que evite los problemas de la incertidumbre. Un ejemplo de este tipo de sistemas es CAPRA [Verdejo, 93], [Garijo et al., 87]. CAPRA es un sistema tutorial que instruye sobre aspectos elementales de la especificación de programas a



estudiantes sin conocimientos de programación. Para ello utiliza modelos individualizados de los estudiantes que incorporan gran cantidad de información sobre el comportamiento de los mismos durante el aprendizaje. Estos modelos están representados como modelos *overlay* enriquecidos con información del proceso de aprendizaje de un nuevo concepto como elementos relacionados ya conocidos, nivel de adquisición del concepto, explicaciones dadas, ejemplos utilizados, problemas resueltos y errores detectados.

JERARQUÍA DE CONCEPTOS DEL SISTEMA	MODELO DE USUARIO	
	Estado de conocimiento del usuario	Certeza en la creencia
Proceso infeccioso	conocido	100
Infección en la cabeza	conocido	100
Infección Subdural	no conocido	100
Otitis	sin información	100
Sinusitis	sin información	100
Meningitis	conocido	100
Meningitis bacteriana	conocido	90
Micobacteria-TB	no conocido	70
Virus	no conocido	90
Meningitis fungal	no conocido	100
Infección micótica	no conocido	100
Encefalitis	no conocido	90
Infección de la piel	conocido	100

Figura 3.15: Ejemplo de modelo overlay

Otro ejemplo de sistema que utiliza este tipo de modelos es UMFE [Sleeman, 85], el cual proporciona explicaciones del funcionamiento de un sistema experto adaptadas al nivel de experiencia del usuario que plantea las preguntas. Para ello se asegura de que en la explicación sólo se utilicen conceptos que el usuario ha dicho que conoce o que el sistema ha asumido que conoce. La técnica de cubrimiento utilizada asocia a cada concepto de la base de conocimiento del sistema un parámetro que indica si el concepto es conocido/desconocido por el usuario o si el sistema no tiene información al respecto, y otro parámetro que indica el grado de certeza que el sistema asigna a esa creencia. Las inferencias sobre los conceptos que el usuario conoce o desconoce están guiadas por reglas deductivas del tipo: *A implica B, C y D*, que permite deducir que si el concepto A es conocido o desconocido entonces los conceptos asociados B, C y D son también conocidos o desconocidos respectivamente. Sin embargo, este tipo de inferencias podría ser poco realista en muchos casos ya que presume una progresión muy rígida en el aprendizaje de conceptos y, no permite representar el hecho de que B, C y D podrían contribuir a A de diferentes maneras, o que algunos de ellos podrían ser



necesarios pero no suficientes para aprender A. La figura 3.15 muestra un ejemplo de modelo *overlay* inspirado en el sistema UMFE.

Una excepción a estos modelos sin capacidad de predicción es el modelo de cubrimiento jerárquico de USCSH [Matthews, Biswas, 86], que tiene una limitada capacidad de predicción. USCSH predice qué conocimiento tiene un usuario sobre nodos padres en la jerarquía mediante una suma ponderada de su nivel de conocimiento de los nodos hijos, lo cual permite hacer predicción sólo sobre conceptos muy relacionados.

### Estereotipos

Un estereotipo es una colección de atributos, usualmente con un grado de certeza asociado, que aparece a menudo en una categoría de usuarios. El uso de estereotipos fue introducido por [Rich 79, 83, 89] en el sistema GRUNDY, en lo que se consideró el inicio de la modelización de usuarios como un área de investigación. GRUNDY utiliza estereotipos para caracterizar diferentes rasgos de la personalidad de sus usuarios y, en base a éstos, sugerirles libros de lectura. La identificación del estereotipo o estereotipos asociados a un usuario se establece tras una serie de preguntas al inicio de la sesión en las que el usuario se describe a sí mismo. Con sus respuestas se activan una serie de disparadores (*triggers*) de los estereotipos que hacen que algunos de ellos pasen a estar activos. Sin embargo, la identificación de un usuario como “principiante” no siempre coincide con la idea del sistema de un “principiante”, por lo que este tipo de estereotipos son útiles para modelizar rasgos de personalidad pero no sirven para modelizar el conocimiento de un usuario sobre el dominio de la aplicación, es decir, con ellos no se puede inferir qué es lo que un usuario sabe.

El estereotipo o estereotipos activados se van adaptando a las características del usuario, descubiertas durante la consulta, mediante un proceso de inferencia que modifica los valores de certeza de los atributos de acuerdo con una reglas predefinidas. El modelo de las características del usuario se establece combinando los distintos atributos o facetas proporcionados por cada uno de los estereotipos activados por ese usuario. Rich evidencia la necesidad de propagar los cambios en un atributo a todos los atributos que dependen de él utilizando sistemas de mantenimiento de la verdad, sin embargo en GRUNDY se adaptan los valores de certeza de atributos individuales pero no



se propagan estas modificaciones a lo largo del modelo de usuario. La figura 3.15 muestra un ejemplo del tipo de representación de estereotipos utilizada en GRUNDY para caracterizar a una persona feminista.

RASGOS DE LA PERSONALIDAD	VALOR	CERTEZA EN LA CREENCIA
Mujer	3	700
Educación	3	800
Tolerancia al sexo	5	700
Perseverancia	3	600
Independencia	3	600
Devoción religiosa	-5	800
Inclinación política	liberal	700

Figura 3.15: Representación de estereotipos en GRUNDY

Otro ejemplo de uso de estereotipos es el del sistema MAP [Macmillan, 83]. Este sistema hace que el usuario identifique el estereotipo al que pertenece mediante una larga sesión de preguntas y respuestas con las que el usuario llega a especificar un modelo completo de cómo interactúa con un sistema operativo. Sin embargo, este sistema es demasiado costoso en tiempo y sólo resulta aceptable cuando se utiliza en tareas como interfaces de sistemas operativos inteligentes, para los que fue diseñado MAP, ya que en estos casos los beneficios se obtienen a lo largo del tiempo. Por tanto, este tipo de enfoque no es admisible en sistemas de consulta donde las sesiones son muy cortas y los usuarios cambian frecuentemente.

[Chin, 89] en su sistema KNOME categoriza los estereotipos de usuarios en cuatro clases de acuerdo con su nivel de conocimiento: novatos, principiantes, intermedios y expertos. La información que ha de presentarse al usuario también es etiquetada de acuerdo con el nivel de dificultad en simple, intermedia, compleja o muy compleja. Las inferencias sobre el conocimiento que posee el usuario se representan como relaciones entre estereotipos de usuario y niveles de dificultad, como se muestra en la figura 3.16. Con este sistema de estereotipos dobles puede predecirse cuál es probablemente el conocimiento del usuario partiendo de una información inicial limitada.

La revisión de la imagen por defecto de un usuario se hace ajustando su categoría y mediante inferencias del conocimiento del usuario sobre hechos particulares. Este proceso de revisión se controla con un sistema de reglas que maneja la incertidumbre con criterios de lógica borrosa. El problema de



KNOME es que no incluye demasiadas reglas explícitas sobre relaciones entre conceptos, es decir, no puede deducir conceptos que el usuario probablemente conozca basándose en conceptos que ya se sabe que conoce.

ESTEREOTIPO DE USUARIO	NIVEL DE DIFICULTAD DEL CONOCIMIENTO			
	simple	medio	complejo	muy complejo
experto	todo	todo	gran parte	-
intermedio	todo	gran parte	poco	-
principiante	gran parte	poco	nada	-
nuevo	poco	nada	nada	nada

Figura 3.16: Representación del nivel de conocimiento de un usuario en KNOME

Otro tipo de representación con estereotipos interesante es la que plantean sistemas como SMMS (*Student Model Maintenance System*) [Huang et al., 91] y EPI-UMOD [De Rosis et al., 92]. En SMMS el conocimiento estereotípico se representa como un grafo dirigido acíclico donde cada nodo representa un subdominio, que incluye varios conceptos o técnicas, y los enlaces representan relaciones entre subdominios. Entonces, el conocimiento por defecto se asocia con conocimiento deductivo que se adquiere mediante un proceso de inferencia puramente lógico, con lo que no hay incertidumbre sobre las creencias en los hechos que un estudiante conoce.

STEREOTYPE'S NAME: Hospital doctor
TRIGGERS
university degree : medicine
postgraduate training in epidemiology: one week course
postgraduate training in statistics : none
job: hospital
years of experience less than three
speciality: cancer treatment
CUERPO
(dichotomic_variable, 0.65)
(risk_factor, 0.72)
(reference_population, 0.47)
(relative_risk, 0.67)
(confounding, 0.67)
(odds_ratio, 0.49)
(adjustment, 0.47 )

Figura 3.17: Ejemplo de esterotipo de usuario en EPI-UMOD  
(de [De Rosis et al., 92])

EPI-UMOD combina la representación basada en estereotipos con una representación explícita de relaciones entre atributos. Los estereotipos de EPI-



UMOD están compuestos de tres partes: el conjunto de atributos que caracterizan al usuario y activan el estereotipo (*triggers section*), el conjunto de atributos que definen conceptos del dominio junto con una medida de certeza asociada (*body section*) y, un componente de inferencia que es un modelo de la evolución del aprendizaje de conceptos representado con una red de creencias.

Las relaciones entre atributos representadas en la red de creencias asocian a los nodos raíz probabilidades iniciales, determinadas por el *body section*, y probabilidades condicionadas a los demás nodos que se van modificando y propagando a medida que se adquieren nuevas evidencias sobre el conocimiento del usuario. Por ejemplo, la figura 3.18 muestra parte de la red de creencias asociada al estereotipo descrito en la figura 3.17.

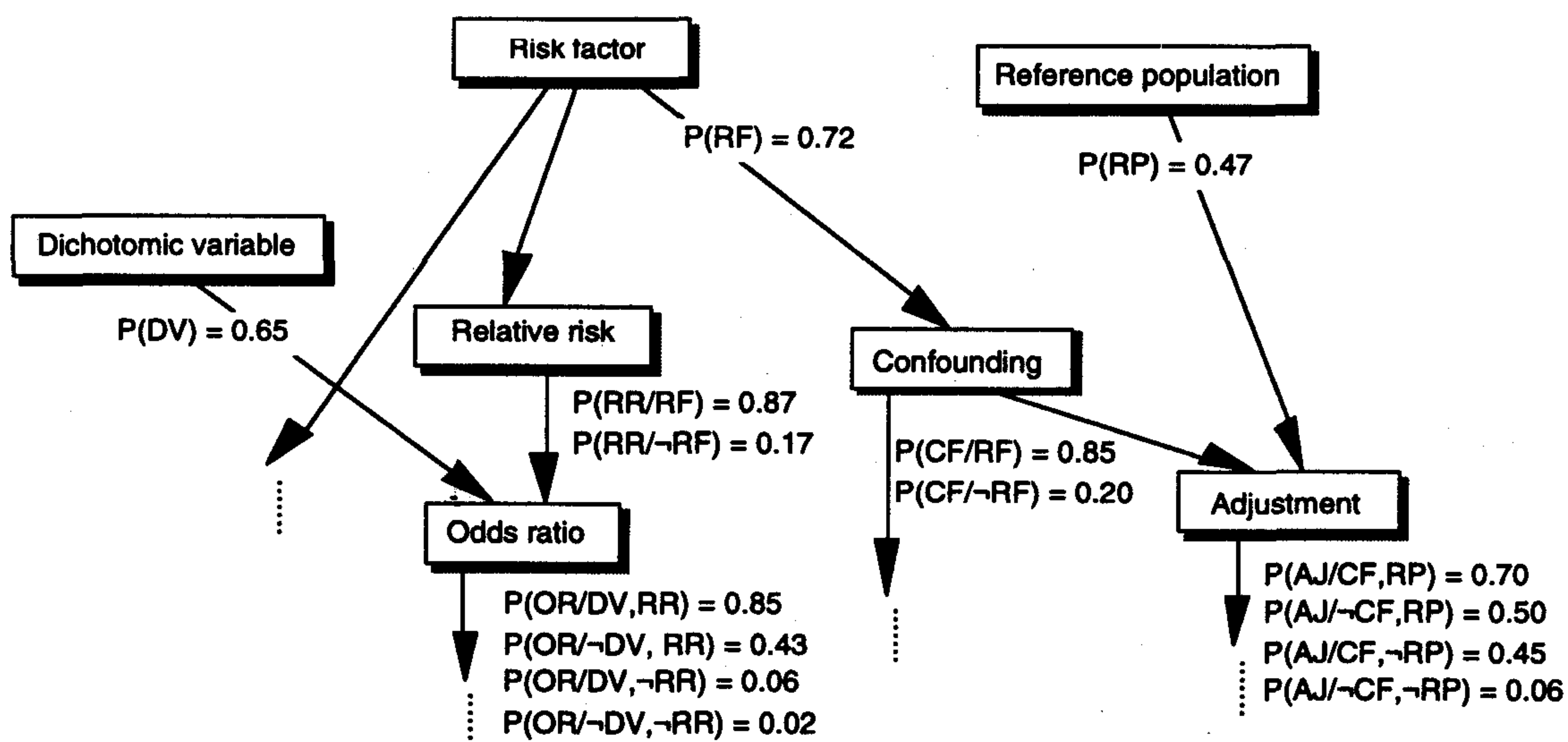


Figura 3.18: Red de creencias asociada a un estereotipo en EPI-UMOD  
(de [De Rosis et al., 92])

Entonces, la precisión del modelo se incrementa mediante preguntas al usuario respecto a conceptos críticos y el uso de información sobre relaciones entre atributos para revisar la incertidumbre sobre todos los valores de los atributos. El principal problema de EPI-UMOD es que la identificación de un usuario con un estereotipo no se modifica una vez fijada, por lo que si la elección ha sido errónea todas las deducciones posteriores sobre lo que el usuario conoce o no lo serán también.



### Técnica de particiones

Otra forma de representar creencias y objetivos es con la *técnica de particiones* (*partition approach*), que puede utilizarse con todas las formas de representación del conocimiento basadas en conceptos [Cohen, 78]. La idea es mantener particiones separadas que recojan las creencias del sistema sobre el dominio, los objetivos del sistema, las hipótesis del sistema sobre las creencias del usuario sobre el dominio, las hipótesis del sistema sobre los objetivos del usuario, las hipótesis del sistema sobre las hipótesis del usuario sobre las creencias del sistema sobre el dominio, etc. Para representar los contenidos de estas creencias y objetivos se utilizan formalismos estandar de representación de conceptos (redes, cálculo de predicados,...). Un ejemplo de esta representación para la creencia y objetivo siguientes: *El usuario cree que Pedro quiere a Ana y desea que Ana quiera a Luis*, es:

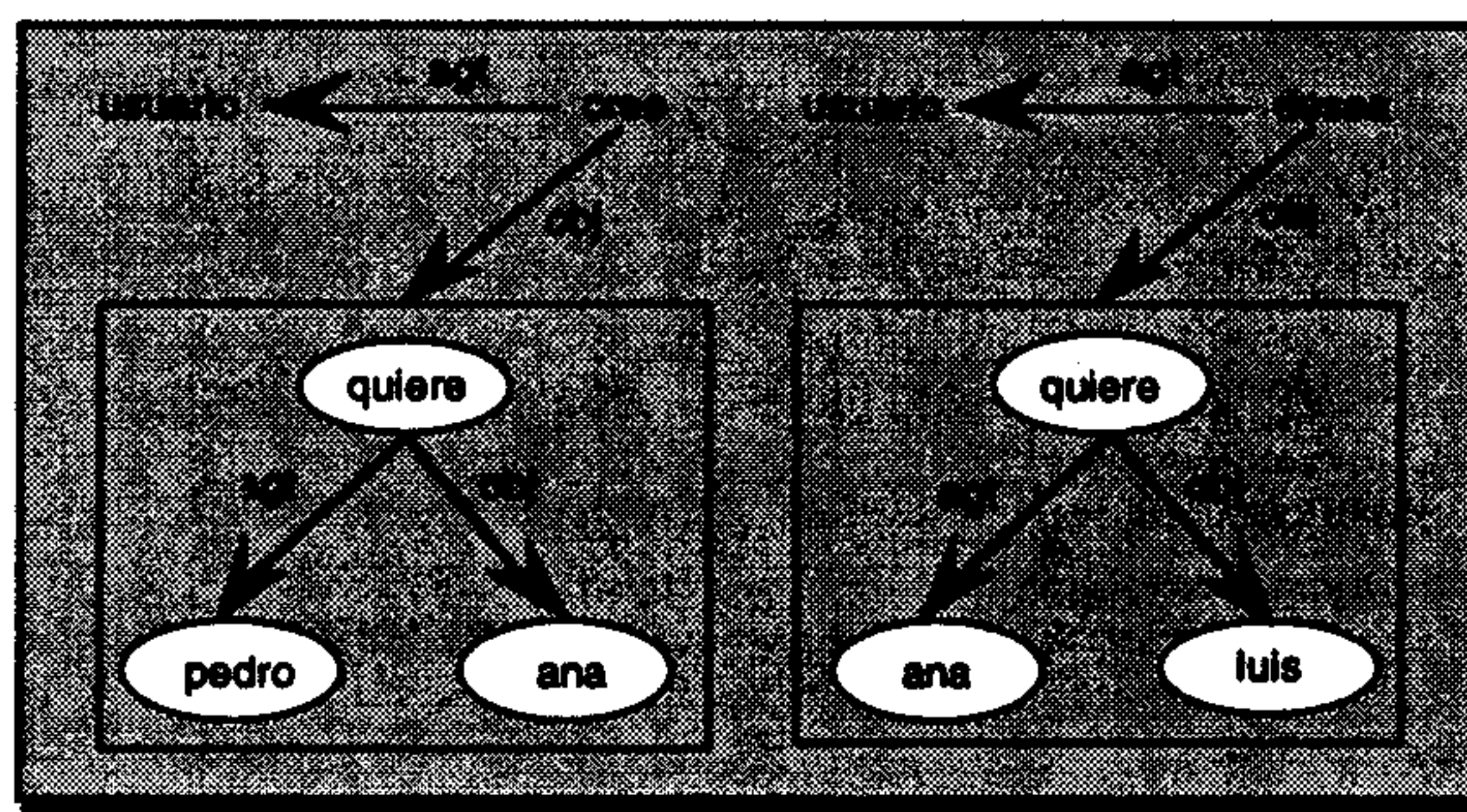


Figura 3.19: Representación con la técnica de particiones

Un sistema interesante que combina el uso de este mecanismo, incluyendo la posibilidad de herencia, con el de los estereotipos es el BGP-MS (*Belief, Goal and Plan Maintenance System*) [Kobsa, 90], que en la actualidad se ha convertido en el *shell* de modelización de usuarios más ampliamente utilizado [Kobsa et al., 94], [Kobsa, Pohl, 95] que se encuentra disponible en la red vía TCP/IP [Fink et al., 97], [Pohl, Höhle, 97]. Este sistema divide el conocimiento en particiones de conceptos que se presume se aprenden a la vez, organizadas jerárquicamente y con posibilidad de herencia, y cuyo patrón de aprendizaje se representa como relaciones entre las particiones. De esta manera se pueden predefinir posibles patrones de aprendizaje definiendo los tipos de relaciones que podrían mantenerse entre particiones.

Las particiones pueden ser de tres tipos: (i) las que contienen el conocimiento sobre modelos individuales de usuario, (ii) las que contienen el conocimiento



de grupos de usuarios, representado como una jerarquía de estereotipos, y (iii) la que contiene el conocimiento del dominio. Cuando el sistema recibe información sobre el conocimiento o desconocimiento del usuario respecto a determinados conceptos hace dos cosas: (i) guarda esta información en dos particiones del modelo individual del usuario: la que contiene lo que sabe y la que contiene lo que no sabe, y (ii) usa esa información para analizar si se verifican las condiciones de activación/desactivación de los estereotipos en intervalos regulares de tiempo. Como resultado de esto, se añadirán o eliminarán enlaces de herencia entre particiones de estereotipos y la partición del modelo individual del usuario en el que reside el conocimiento sobre lo que sabe.

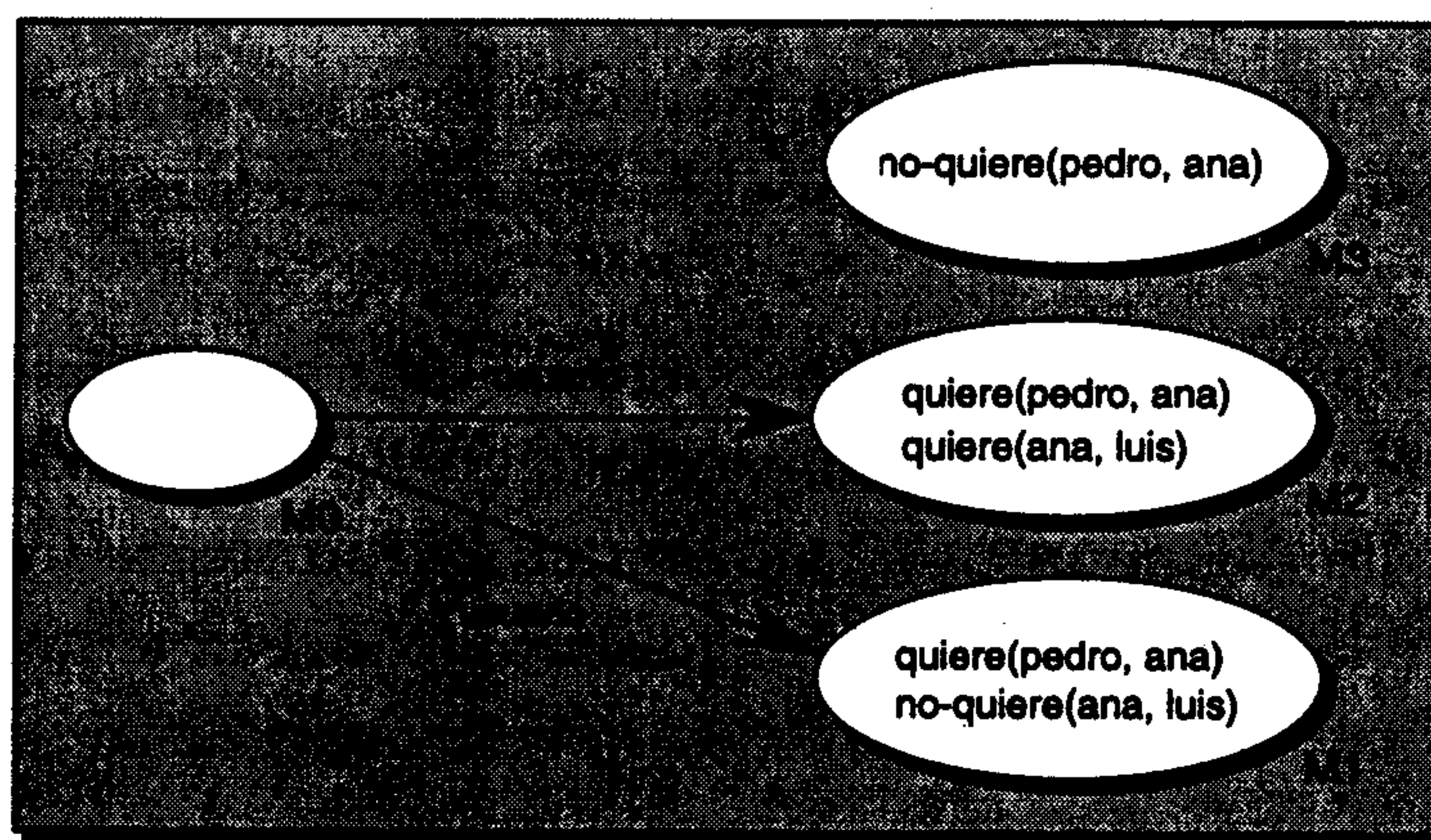


Figura 3.20: Representación con mundos posibles

Por otro lado, dentro del campo de la lógica se ha utilizado la lógica modal para representar creencias y objetivos, definiendo la llamada *semántica de mundos posibles*. En esta representación las fórmulas modales son traducidas a sentencias de un mundo posible sobre las que pueden aplicarse técnicas de deducción estándar. Los operadores modales *cree* y *quiere* se traducen a relaciones de accesibilidad entre mundos posibles. Entonces, el usuario sabe que un hecho *H* es cierto si lo es en todos los mundos compatibles con el que él conoce. Por ejemplo, la representación de la creencia: *El usuario sabe que Pedro quiere a Ana y no está seguro de si Ana quiere a Luis*, es la que muestra la figura 3.20 donde M0 es el mundo con el que se identifica el usuario. Aquí el usuario conoce el hecho *quiere(pedro, ana)* pero duda respecto al hecho *quiere(ana, luis)* porque hay un mundo que le es accesible desde M0 en el que la negación de *quiere(ana, luis)* es cierta (M1). El problema fundamental de esta técnica es su limitada capacidad expresiva para los problemas de representación de modelos de usuario.



## 3.5 Diseño de Interfaces de Usuario Basado en Modelos

Uno de los aspectos que más ha influido en la mejora de la usabilidad de los sistemas informáticos ha sido la evolución en el estilo de interacción con los usuarios, evolución de la que se ha dado una muestra en las secciones anteriores, y que ha pasado de los interfaces basados en comandos crípticos, con los que sólo unos pocos estaban familiarizados, a los modernos interfaces multimedia/multimodo en los que la interacción se facilita al disponer de varios canales de comunicación. Las etapas intermedias se han caracterizado por el uso generalizado de pantallas gráficas y ratones que permitieron la introducción en un principio de sistemas basados en menús, que reducían la necesidad de memorizar comandos, hasta llegar a la metáfora de interacción dominante en estos momentos que son los interfaces de manipulación directa.

Sin embargo, la introducción de estos potentes recursos de presentación en los interfaces ha tenido un coste en la construcción de los mismos que ha generado la necesidad de disponer de herramientas software que ayuden al programador en las distintas etapas del desarrollo de la interfaz. La respuesta a esta necesidad han sido los *Sistemas de Gestión de Interfaces de Usuario* (UIMS en inglés). Un UIMS, en sentido amplio, es una aplicación de software interactivo de alto nivel que facilita el desarrollo de interfaces de usuario en todas las etapas de su construcción: diseño, implementación, ejecución y mantenimiento, y que juega un papel análogo al de las herramientas CASE en la ingeniería de software.

Aunque todavía no existen UIMS que cubran satisfactoriamente todas las etapas del desarrollo de un interfaz, lo cual ha hecho que numerosos autores los denominen UIDS (*User Interface Development Environment*), en todos ellos pueden observarse un conjunto de objetivos comunes, [Hix, 90], [Dix et al., 93, 98], entre los que se incluyen:

- proporcionar un conjunto de técnicas para diseñar, implementar y evaluar un entorno de comunicación interactivo en tiempo de ejecución.
- mejorar la productividad del desarrollador de interfaces mediante su alejamiento de las tareas de codificación.

- involucrar a expertos en factores humanos y usuarios finales en todas las etapas de la construcción del interfaz con el fin de obtener productos con mayor nivel de usabilidad.

Pero sobre todo, el objetivo fundamental de los UIMS es manejar una arquitectura conceptual sobre la estructura del sistema interactivo que se concentre en la separación entre la semántica de la aplicación y la presentación, de manera que ésta se ocupa de generar la información y el UIMS se ocupa de la forma en que dicha información ha de presentarse al usuario. Los argumentos que apoyan la separación entre la semántica de una aplicación y el interfaz proporcionado al usuario para hacer uso de esa semántica son numerosos [Dix et al., 93], y entre ellos destacan la *portabilidad*, *reusabilidad*, la posibilidad de desarrollar *múltiples interfaces* para una aplicación que aumenten su capacidad interactiva, y la *personalización* del interfaz para aumentar su efectividad sin alterar la aplicación. De esta manera, además, quedan caracterizados conceptualmente los tres componentes de un sistema interactivo: la aplicación, la presentación y el control del diálogo.

Por otra parte, [Puerta et al., 94] identifica una serie de deficiencias en las herramientas utilizadas tradicionalmente en el desarrollo de interfaces, fundamentalmente debidas a que el desarrollador se ve obligado a trabajar a muy bajo nivel de abstracción. La solución que plantea es desarrollar UIMS que puedan ofrecer a los desarrolladores un *modelo declarativo de interfaz* que permita manipular todas las facetas de la construcción de un interfaz a un alto nivel de abstracción. UIMS de este tipo a destacar son UIDE [Foley et al., 88, 91], HUMANOID [Szekely et al., 92, 93], [Frank, Szekely, 98], ADEPT [Wilson et al., 93, 94], MECANO [Puerta et al., 94], MOBI-D [Puerta, Maulsby, 97], [Puerta, 97].

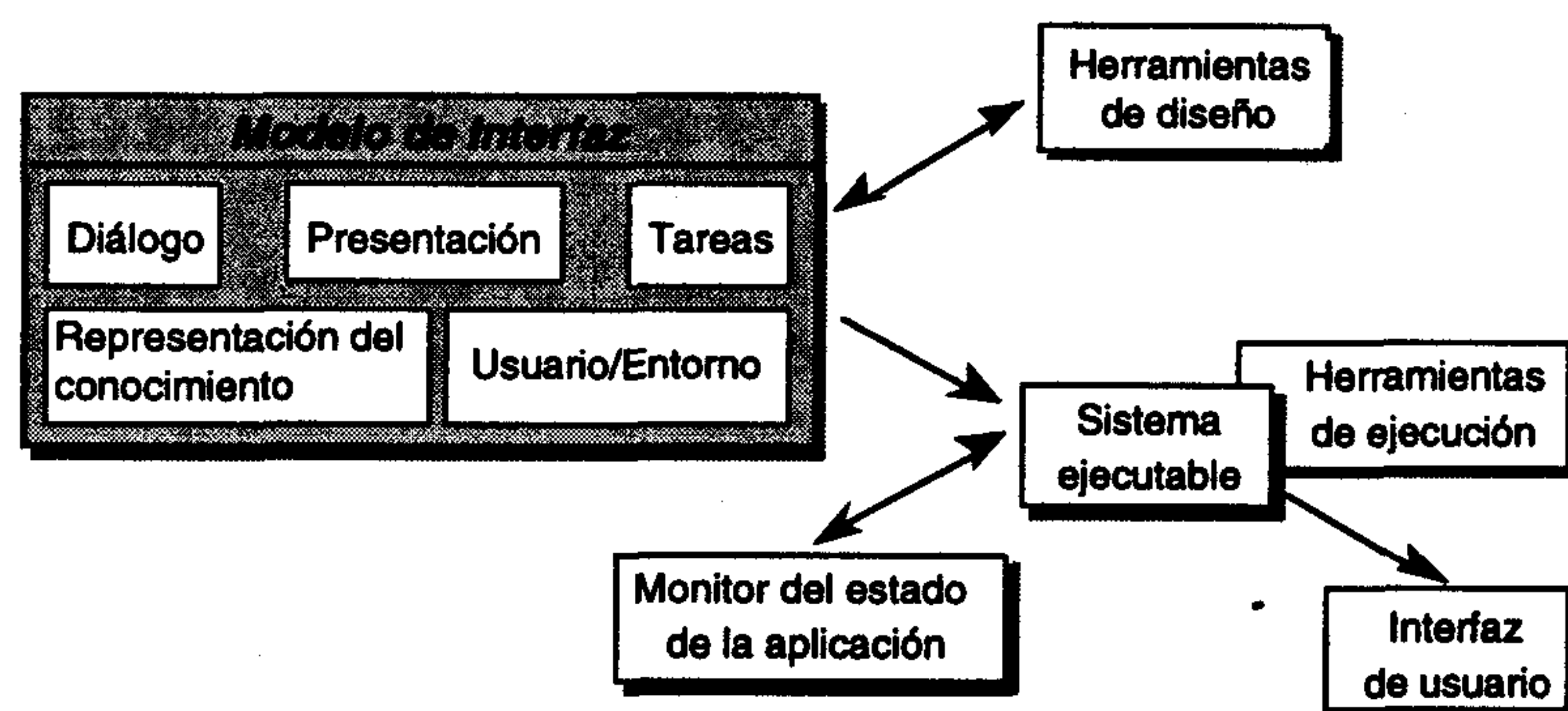
Como el modelo de interfaz que emplean estos sistemas cubre todos los aspectos del diseño y desarrollo del interfaz puede ser usado como un depósito central de conocimiento que las distintas herramientas pueden utilizar para llevar a cabo las funciones relacionadas con la construcción del interfaz. El tipo de conocimiento que considera incluye: información sobre las tareas que el usuario realizará con la aplicación, objetos del dominio y acciones sobre ellos, comandos que el usuario puede utilizar, presentación y comportamiento de la interfaz, y características de los usuarios a tener en cuenta en la presentación de la información. Las ventajas que proporciona el uso de este paradigma son las



que ofrece la definición de cualquier tipo de diseño a niveles superiores de abstracción, que en el caso del diseño de interfaces se traduce en que:

- facilita la construcción y modificación del diseño del interfaz al utilizar representaciones a alto nivel;
- facilita la reusabilidad del modelo de interfaz;
- mejora la usabilidad de la aplicación que utilice el interfaz así diseñado;
- permite extender las funcionalidades del interfaz en tiempo de ejecución con, por ejemplo, ayudas, explicaciones o caracterización de usuarios;
- soporta la generación automática de interfaces.

La arquitectura de un sistema de desarrollo de interfaces basados en modelos (ver figura 3.21) está compuesta de varios componentes que interactúan de la forma siguiente: Los desarrolladores del interfaz de usuario pasan de una descripción genérica de interfaces de usuario - el modelo de interfaz - a una descripción específica de un interfaz particular - una instancia del modelo de interfaz - utilizando herramientas de diseño. El entorno de desarrollo y ejecución toma la instancia del modelo y la implementa como un interfaz ejecutable. Las herramientas de ejecución dan soporte al uso del interfaz, con funciones como ayudas, controlando el estado de la interacción mediante un monitor del estado de la aplicación.



*Figura 3.21: Arquitectura básica de un UIMS basado en modelos  
(adaptado de[Puerta et al., 94])*

En resumen, el cambio radical experimentado por los interfaces de usuario en los últimos años ha girado en torno a dos objetivos fundamentales:

- obtener interfaces que aumenten el grado de usabilidad del sistema, es decir, que ofrezcan al usuario un entorno de trabajo comprensible y confortable, y
- obtener entornos que faciliten al programador la construcción de interfaces cada vez más complejos en cuyo diseño puedan intervenir expertos en factores humanos y los propios usuarios finales.

## 3.6 Conclusiones

En este capítulo se han presentado diversas soluciones de interacción usuario-sistema desde la perspectiva de los interfaces de usuario inteligentes. Estas propuestas tienen como común denominador que se han desarrollado a partir del análisis de la comunicación humana, por lo que gran parte de los esfuerzos dedicados en este campo se han orientado al desarrollo de técnicas que permitan transmitir información en lenguaje natural.

La transición entre la primera y segunda generación de generadores de presentaciones se produce en el momento en el que estos sistemas empiezan a incorporar y utilizar representaciones explícitas tanto de la intención del sistema cuando se comunica con el usuario como del efecto que éste espera provocar en ese usuario tras recibir la información, lo que con el tiempo ha llevado a su vez al desarrollo de las técnicas de modelización de usuarios. Así, los interfaces de usuario pasan de limitarse a organizar en un texto coherente la información a presentar, a elaborar este texto guiados por criterios de efectividad, es decir, persiguiendo un impacto determinado en el estado de conocimiento del usuario.

Con la aparición de los dispositivos multimedia se dio un salto cualitativo en el nivel de comunicación usuario-sistema ya que al diversificar los canales de transmisión de información se podían aprovechar las capacidades expresivas de cada uno de los medios, haciendo que las carencias de unos las supliesen las potencialidades de los otros. Sin embargo, este desarrollo de los interfaces de usuario se ha hecho sin analizar la naturaleza de la información que se transmite a través de los mismos, es decir, cuál es el origen de esa información,



cómo se ha obtenido y por qué se ha de presentar esa información y no otra; lo cual puede redundar en una infraexplotación de sus posibilidades.

## 4. Análisis

---

Como resultado de la revisión de las propuestas anteriores se identifica un avance significativo en la calidad de la comunicación entre sistemas y usuarios alcanzado desde dos líneas de trabajo distintas y distantes:

- la que ofrecen los sistemas basados en el conocimiento, que buscan acercar al usuario al mundo de la aplicación mediante una aproximación al modelo de comprensión del mundo que posee el usuario, y
- la de los interfaces de usuario, que intentan acercar el sistema al usuario mediante la explotación de los medios de comunicación del sistema con el fin de enriquecer su capacidad expresiva y aumentar las posibilidades de éxito en la recepción del mensaje por parte del usuario.

A continuación se presenta un resumen de las propuestas revisadas y después se realizan algunas críticas que dan pie a la solución planteada en esta tesis.

### 4.1 Resumen de las Propuestas

En el primer caso el éxito de la comunicación radica en la capacidad del ingeniero del conocimiento para desarrollar un modelo de organización y representación del conocimiento de un dominio problema que sea similar al que posee el experto (esto suponiendo que los expertos humanos tienen su conocimiento estructurado de alguna forma) o al menos que éste sea capaz de reconocer, comprender y modificar. Así, este tipo de sistemas suplen sus carencias en expresividad con su *apertura* hacia el usuario, haciendo visibles su conocimiento y su modo de razonar con él, y proporcionando explicaciones de la forma en la que alcanzan sus conclusiones. En concreto, las características comunes de las soluciones presentadas en este sentido son las siguientes:



- conciben una aplicación como un agente inteligente, en el sentido de Newell, para el que se describen en primer lugar el conocimiento de que dispone y la forma en que lo usa, antes de preocuparse por aspectos relativos a su representación computable;
- manejan un modelo de estructuración del conocimiento jerárquico y modular, organizado en torno a módulos especializados en la realización de diferentes tipos de tareas que se agrupan, formando una arquitectura, en función del rol que desempeñan en el proceso de resolución de problemas;
- la descripción a alto nivel del conocimiento y funcionalidades de una aplicación junto con el modelo de organización empleado produce estructuras abstractas de resolución de problemas que son aplicables en diferentes dominios mediante la reutilización y particularización, total o parcial, de estas estructuras y sus módulos especializados componentes.
- todas ellas han dado lugar al desarrollo de herramientas para la construcción de aplicaciones basadas en el uso de piezas software preprogramadas que realizan tareas elementales y que constituyen el soporte computacional de los modelos estructurados de conocimiento.

Un resumen de las características más relevantes de estas propuestas sería el siguiente:

- La perspectiva a nivel de conocimiento propuesta por Newell presenta dos características fundamentales:
  - (i) reduce la distancia entre las formas de resolver problemas de los expertos de un dominio y las formas tradicionales de representación del conocimiento de esos expertos, al plantear la descripción del conocimiento y el modo en que debe usarse sin preocuparse de su representación computable.
  - (ii) modifica la metodología de trabajo de los ingenieros del conocimiento por la que se impone la reflexión sobre la forma de entender un problema como paso previo al diseño informático.

- Las tareas genéricas de Chandrasekaran constituyen una propuesta sobre modularización y estructuración de los sistemas basados en el conocimiento, al presentarse como piezas básicas sobre las que construir arquitecturas complejas. Con el concepto de tarea genérica, además de indicar la forma en que debe estructurarse el conocimiento, se pueden representar explícitamente estructuras de inferencia, lo cual es muy útil para mejorar la explicación y mantenimiento del conocimiento.
- De forma análoga a las tareas genéricas, los métodos *role-limiting* resultan muy útiles para construir sistemas basados en el conocimiento al ofrecer la posibilidad de diseñar arquitecturas especializadas en la resolución de clases de problemas, que pueden aplicarse a diferentes dominios mediante la especificación del contenido de sus bases de conocimiento. Así mismo, la aplicación de estos métodos, al igual que las tareas genéricas, permite dar un paso más en el acercamiento usuario-sistema al poder justificar el sistema sus conclusiones en base al uso que hace del conocimiento en cada etapa de razonamiento.
- Los modelos conceptuales de KADS y Steels son propuestas sobre cómo abordar la etapa de análisis de un sistema a nivel de conocimiento. Presentan grandes similitudes al contar ambas con la jerarquía de tareas como núcleo de la organización del conocimiento, que tiene en su nivel inferior las acciones de inferencia elementales que en el caso de KADS se corresponden con las fuentes de conocimiento que procesan metaclases, mientras que en el caso de la propuesta de Steels son los métodos solución que procesan modelos. Como diferencia cabe distinguir los niveles estratégico y de dominio de KADS, que permiten dar más detalles sobre el conocimiento del sistema que en la propuesta de Steels.
- El modelo de estructuración del conocimiento basado en unidades cognitivas además de la perspectiva funcional de tareas ofrecida también por las propuestas de KADS y Steels, introduce la perspectiva de áreas de conocimiento de un sistema como una visión adicional de la organización del conocimiento que está más próxima al planteamiento de agente inteligente de Newell y que supone un valor añadido de cara a la comprensión del modelo de resolución de problemas del sistema.



Con respecto al conocimiento de control de estas estructuras de conocimiento, cabe hacer dos distinciones: (i) aquel que determina el orden en el que deben realizarse las subtareas asociadas a cada tarea, y (ii) aquel que decide, ante un conjunto de métodos alternativos, cuál de ellos debe llevar a cabo una tarea. Con relación al primero, las propuestas presentadas utilizan las siguientes soluciones:

- KREST emplea diagramas de estados y transiciones etiquetadas con estados de control asociados de forma local a cada tarea. Ante el estado de control obtenido de la realización de una tarea, queda determinada la transición a considerar y por tanto la siguiente tarea que debe realizarse.
- PROTÉGÉ-II utiliza una agenda global en la que representa las tareas que quedan por ejecutar y de la que elige la tarea que debe realizarse en cada momento, de acuerdo con criterios generales de control.
- KSM asocia localmente a cada tarea un conjunto de reglas de producción que basándose en los valores de estados de control determinan una secuencia de llamadas a determinadas funcionalidades de las unidades cognitivas del modelo.

Por otra parte, con respecto al conocimiento de selección de métodos, las soluciones empleadas son las siguientes:

- TIPS explota las posibilidades de control flexible de las estructuras de tareas genéricas asociando localmente a las tareas que disponen de métodos alternativos un mecanismo espónsor-selector. Cada método tiene asociado un espónsor que evalúa la adecuación del método en función del estado de resolución de problemas. El selector decide, a la vista de las medidas de adecuación ofrecidas por los selectores y el estado de resolución de problemas, el método que ha de realizar la tarea.
- MODEL-K instrumentaliza el nivel estratégico de KADS mediante un modelo de conocimiento reflexivo global gestionado por un planificador. Las tareas metanivel controladas por el planificador están orientadas a la observación de aspectos generales del proceso de resolución de problemas (como analizar la dificultad de un problema, predecir los recursos necesarios o identificar puntos problemáticos del razonamiento)

ya que la decisión del planificador sólo afecta a las tareas de más alto nivel, cuya realización sigue una secuencia fija de pasos.

- TroTelC asocia explícitamente a cada tarea la lista de métodos que puede utilizar y para cada uno de ellos sus condiciones de aplicabilidad, referidas a la disponibilidad de determinado conocimiento del dominio. La elección entre diferentes métodos aplicables se toma con conocimiento implícito y por tanto no modificable.

En el caso de los interfaces de usuario, es el sistema el que hace un esfuerzo por comunicar sus resultados al usuario en forma comprensible. Para lograr este objetivo, las soluciones presentadas tienen en común que consideran a los predicados retóricos como las piezas constitutivas de cualquier texto en lenguaje natural; sin embargo, estas soluciones emplean dos tipos de modelos de estructuración de estos predicados:

- el modelo basado en relaciones retóricas entre las distintas partes de un texto, con el que no es posible expresar una organización jerárquica del texto o precedencias entre las piezas componentes; aplicado en las soluciones basadas en esquemas o plantillas de McKeown, Hovy y en los operadores de Moore-Paris.
- el modelo basado en actos comunicativos, organizados jerárquicamente mostrando la descomposición de actos comunicativos de alto nivel en otros más básicos hasta llegar al nivel de los actos locutivos. En este caso, el modelo se centra en las propiedades de cada acto comunicativo sin expresar las relaciones entre las distintas partes del texto. Este fue aplicado en los operadores de planificación de Maybury.

La calificación de un interfaz de usuario como inteligente proviene de la representación explícita de las intenciones del sistema y el efecto que persigue. Se pasa de la obsesión por obtener una configuración consistente de un texto a la búsqueda de un objetivo comunicativo a través de él. Así, los interfaces de usuario pasan de limitarse a organizar la información a presentar en un texto coherente, a elaborar este texto guiados por criterios de efectividad, es decir, persiguiendo un impacto determinado en el conocimiento del usuario.



Aunque todas las propuestas presentadas emplean algún tipo de modelo del conocimiento del usuario, estos modelos se gestionan con soluciones muy sencillas:

- el planificador de Moore-Paris utiliza un modelo percedero basado en predicados, cuya vigencia dura lo que dura el proceso de planificación de la presentación, y que en realidad representa el conjunto de objetivos comunicativos que persigue el sistema con la presentación.
- en el caso de Maybury, tanto en su planificador de presentaciones textuales como multimedia, el modelo empleado es muy similar al anterior, con la diferencia de que representa explícitamente el efecto de cada operador sobre el conocimiento del usuario de forma separada de los objetivos retóricos que persigue el operador.
- WIP y COMET no especifican el tipo de modelo de usuario que emplean y en la dinámica de sus soluciones de presentación no parecen tener un papel especialmente relevante. En el caso de WIP, cabe distinguir cierta capacidad de adaptación de la presentación a características o preferencias del usuario, como nivel de conocimiento, idioma elegido y presentación incremental, que el usuario define al inicio de una sesión.

Un aspecto especialmente relevante en las propuestas de interfaces multimedia es la coordinación de los distintos dispositivos multimedia disponibles para elaborar una presentación coherente:

- Maybury expresa el modo de presentar cada pieza de información de forma explícita en la definición de los operadores y en las métricas de preferencia entre ellos, restando así flexibilidad a la generación de las presentaciones pero evitando problemas de coordinación al estar ésta predefinida.
- COMET asocia simultáneamente a todas las piezas de información a presentar el modo en que debe expresarse cada una de ellas. Esta información es accesible por los distintos generadores de presentaciones de manera que disponen de una visión global del contexto de la presentación y pueden controlar sus influencias mutuas.

- WIP asocia el modo de presentación de cada pieza de información de forma incremental, soportado por un mecanismo de transferencia de información entre el planificador del contenido y los generadores de presentaciones, que hace posible la evaluación y revisión del plan global en cada paso de planificación, proporcionando así un grado de flexibilidad en las presentaciones superior al de la solución de Maybury y COMET.

Las herramientas desarrolladas para manejar la complejidad de estos nuevos interfaces, o UIMS, presentan dos características comunes:

- mantienen una clara separación entre la aplicación que proporciona la información y, el interfaz que la muestra y se comunica con el usuario;
- utilizan una descripción abstracta del diseño del interfaz representada en un modelo declarativo.

Estos UIMS aplican las ventajas de la modularidad al diseño de interfaces, identificando por separado el conocimiento relativo al diálogo con el usuario y las tareas que éste puede realizar, la presentación, las características del usuario y del contexto de la interacción.

## 4.2 Problemas Pendientes

A la vista de las soluciones presentadas, se ha observado como problema fundamental que la caracterización de un sistema como interlocutor de un usuario está desequilibrada en el sentido de que:

- desde la perspectiva del modelo de conocimiento, éste sabe cómo resolver problemas en un dominio pero no sabe cómo dirigir esa capacidad hacia los intereses de un usuario;
- desde la perspectiva del interfaz de usuario, éste sabe cómo comunicar información a un usuario pero desconoce la naturaleza de la información que maneja.



Es decir, la integración entre el generador de presentaciones y la aplicación que le surte de información es deficiente, ya que por una parte, los sistemas basados en el conocimiento no ofrecen a sus interfaces de usuario los criterios que han determinado el que se produzca un tipo de respuesta concreto, y por otra parte, los interfaces de usuario inteligentes no hacen partícipe a las aplicaciones que les proporcionan la información que luego presentan del conocimiento que disponen sobre el comportamiento observado en el usuario o el éxito de la comunicación. En este sentido cabe decir que, con las soluciones actuales, si se concibe un sistema complejo diseñado con una arquitectura cognitiva comunicada con el usuario mediante un interfaz de usuario inteligente, tanto el interfaz como el modelo de conocimiento se verán mutuamente como *cajas negras*.

Los aspectos problemáticos, observados en ambas líneas de investigación, que apoyan la conclusión anterior son los siguientes:

⇒ En los Sistemas Inteligentes

- *Utilización de las arquitecturas cognitivas sin considerar al usuario.* Las posibilidades que ofrecen los modelos estructurados de conocimiento que consideran múltiples formas de resolver la misma clase de problemas, tanto en su concepción teórica como en sus realizaciones prácticas, en ningún caso se utilizan teniendo en cuenta aspectos relacionados con el tipo de usuario con el que se interactúa ni el contexto en el que se produce esa interacción. Esto puede observarse en los módulos de razonamiento metanivel de las propuestas presentadas, ya que sólo se apoyan en el análisis del estado de resolución de problemas, tanto a nivel de métodos, observando la disponibilidad del conocimiento del dominio que necesitan, como en el caso de TIPS, a un nivel superior, observando la sintonía entre los métodos seleccionados en etapas anteriores, para decidir el siguiente paso a dar en el proceso de razonamiento.
- *La complejidad de los modelos dinámicos de conocimiento no redundará en mayor usabilidad.* Como consecuencia del problema antes descrito se observa que el esfuerzo dedicado en el diseño y construcción de sistemas inteligentes complejos no aumenta la usabilidad del sistema final al olvidar las necesidades de sus usuarios. La redundancia inherente a los

modelos que disponen de múltiples formas de resolver problemas, y el consiguiente coste de mantenimiento no parece fácilmente justificable bajo los criterios de explotación que se han aplicado en las propuestas presentadas ya que el usuario no percibe de forma clara los beneficios. El incremento perseguido en el nivel de usabilidad de estos sistemas dependería fundamentalmente del nivel de servicio y la capacidad para satisfacer las demandas de información de los usuarios que pudiese proporcionar el sistema; pero las soluciones ofrecidas hasta ahora son ajenas a consideraciones de calidad de interacción.

Además, en estos sistemas, la valoración del grado de adecuación de un conjunto de métodos frente a otros para resolver un mismo tipo de problema se hace de forma individual e independiente, y se expresa en términos absolutos. Pero, el valor de las prestaciones o propiedades de un método no es un concepto absoluto, sino que está condicionado por los valores que otros métodos tengan para esas mismas características. La asignación de medidas de adecuación absolutas hace que la incorporación de nuevos métodos suponga revisar las medidas de adecuación asociadas a los métodos ya existentes, para que los nuevos métodos tomen el valor que les corresponde en los distintos escenarios de resolución de problemas.

- *Reusabilidad limitada por el modelo de interacción.* Los factores descritos anteriormente hacen que la reusabilidad efectiva de los sistemas basados en el conocimiento, facilitada por la disponibilidad de modelos genéricos de resolución de problemas, está condicionada por la presencia en el dominio concreto sobre el que se aplica el modelo genérico, no sólo del mismo tipo de conocimiento sino por un mismo modo de utilizar ese conocimiento.

⇒ En los Interfaces de Usuario Inteligentes

- *La flexibilidad de los interfaces inteligentes no implica adaptabilidad real.* De la flexibilidad alcanzada por los modelos de interacción reflejados en las distintas propuestas presentadas, tanto desde la perspectiva de las arquitecturas cognitivas como del interfaz de usuario, no emerge directamente la capacidad de adaptación del sistema a sus usuarios. Incluso en aquellos casos en los que se utilizan modelos de usuario, el



conocimiento incluido en estos modelos, en el mejor de los casos, refleja lo que el usuario sabe pero no lo que quiere saber, es decir, cuál es el objetivo con el que se comunica con el sistema, por lo que estos interfaces no tienen la capacidad de ajustar su comportamiento a esas actitudes del usuario.

En los siguientes capítulos se desarrolla la propuesta de esta tesis que plantea la posibilidad de hacer confluir las dos líneas de investigación presentadas en los capítulos anteriores, integrando de forma coherente las funcionalidades proporcionadas por ambos elementos, interfaz y aplicación, mediante una arquitectura basada en el conocimiento en la que se integra y articula tanto el conocimiento para gestionar un diálogo inteligente con el usuario, como el conocimiento para generar una gama de respuestas más amplia que la de los sistemas convencionales soportadas por un modelo de interacción avanzado.

## **Parte II: PROPUESTA**





# **5. Interacción Usuario-Sistema Avanzada**

---

En este capítulo y el siguiente se presenta el modelo de interacción usuario-sistema que propone esta tesis para elevar la capacidad comunicativa de los denominados sistemas inteligentes en contextos de operación de toma de decisiones.

Este capítulo está dedicado a presentar la visión de la interacción usuario-sistema que persigue esta tesis y los principios que ha guiado el desarrollo de la misma en los entornos de toma de decisiones, marcando de esta manera las metas a conseguir respecto al nivel de comunicación entre un sistema y sus usuarios.

## **5.1 Planteamiento General**

Como ya se vio en los capítulos anteriores, los avances en la mejora del nivel de comunicación entre sistemas orientados a la resolución de problemas en un dominio y los usuarios de estos sistemas, han seguido dos líneas de trabajo completamente distintas. Por una parte, la investigación en interfaces de usuario inteligentes ha buscado el acercamiento de un sistema a su usuario capacitándolo para generar presentaciones coloristas adaptadas a características del usuario en las que la naturaleza y la forma de obtener la información transmitida no es especialmente relevante. Por otra, se ha intentado atraer al usuario hacia el mundo del sistema mediante la construcción de sistemas inteligentes cuyo modelo de resolución de problemas se asemeja al que emplean los expertos en ese dominio, confiando en que esta similitud y la formulación del conocimiento de soporte en términos comprensibles para los usuarios sea suficiente para alcanzar el éxito en la comunicación sin dedicar mayores esfuerzos a la presentación de las conclusiones.



Sin embargo, ambos modos de interacción presentan deficiencias ya que, en el primer caso, se pueden estar desaprovechando las capacidades expresivas del sistema al transmitir información cuya elaboración no puede ser justificada, haciendo por tanto que el usuario reciba una visión superficial del contenido del mensaje. En el segundo caso, el uso de técnicas básicas de presentación en las que no se tiene en cuenta el tipo de usuario con el que se está comunicando ni el contexto en el que se produce esta comunicación puede hacer que toda esa información generada no sea adecuadamente comprendida por el usuario.

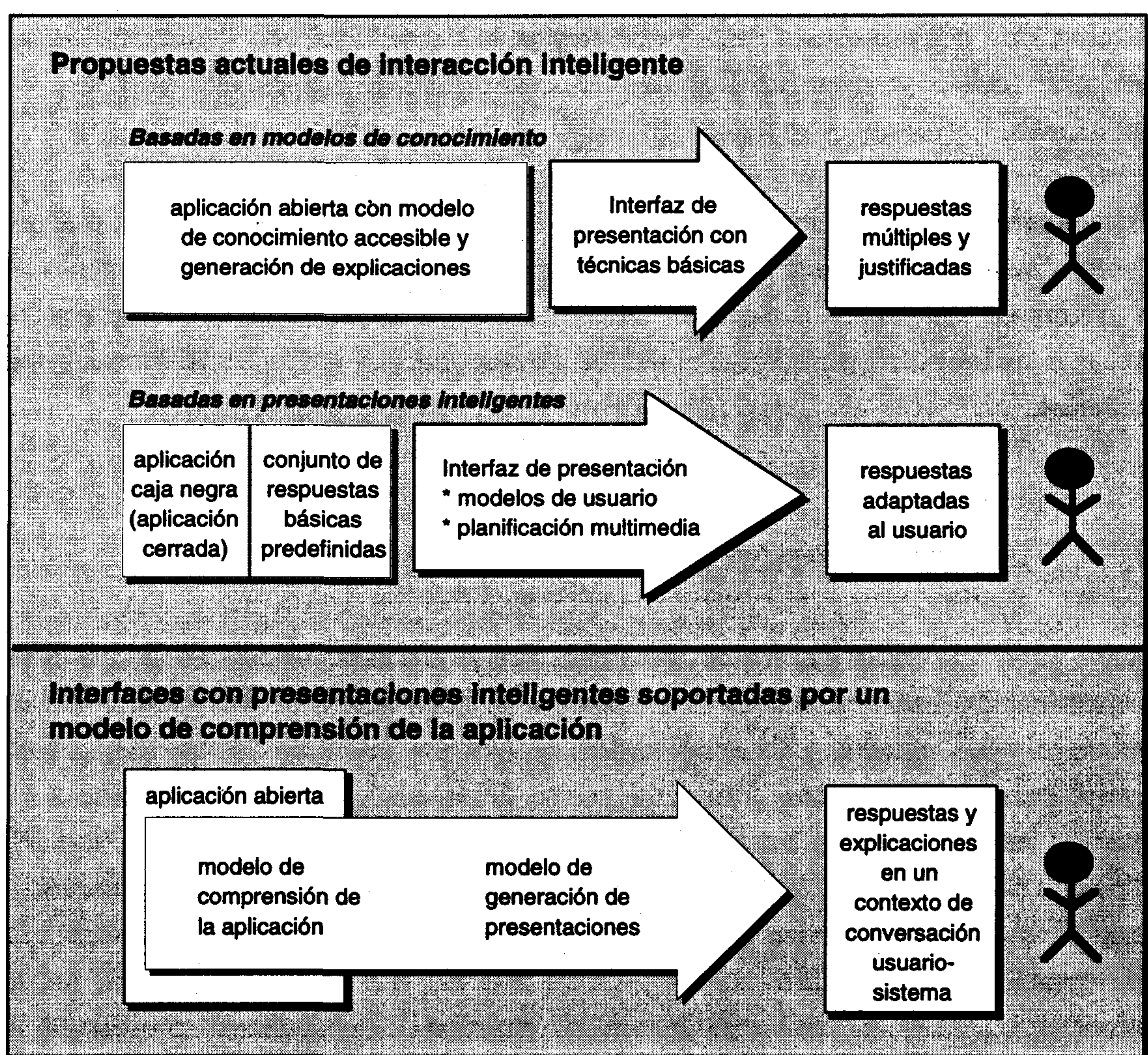


Figura 5.1: Tres visiones de la interacción con el usuario

Esta tesis plantea la necesidad de establecer un nexo de unión entre las soluciones centradas en el interfaz y las centradas en la aplicación para regular el desequilibrio provocado por la división actual entre formas de resolver y



formas de presentar. Se busca conseguir una integración tanto funcional como a lo largo del tiempo entre un modelo de comprensión del dominio problema de una aplicación y un modelo generador de presentaciones que sea coherente con el contenido de la aplicación. Para ello, se plantea un modelo de interacción capaz de mantener sesiones de preguntas y respuestas entre el usuario y el sistema en las que la generación de las respuestas responde no sólo a criterios de disponibilidad del conocimiento necesario para razonar sino también a consideraciones sobre la forma en la que debe obtenerse esa respuesta (ver figura 5.1). Este último aspecto plantea una postura novedosa con respecto a la forma de articular los diálogos usuario-sistema: el tipo de diálogo que se establece entre un usuario y un sistema condiciona la forma en que van a resolverse los problemas que darán respuesta a las demandas del usuario.

Ahora bien, con las miras puestas en alcanzar un mayor nivel de interacción ente usuarios y sistemas, la sola capacidad de generar múltiples respuestas para una misma pregunta no es suficiente. O mejor dicho, la comunicación entre un sistema y sus usuarios se verá realmente enriquecida cuando esa capacidad de razonamiento se explote bajo criterios de calidad de la interacción. En este sentido, se pone de manifiesto la *necesidad de razonar sobre formas de conversar como requisito para decidir las formas de resolver problemas*. Esta postura es radicalmente opuesta a la que se observa en los sistemas existentes, ya que en éstos, el punto de partida es la selección del modo de razonar, para después aplicar mecanismos más o menos sofisticados de presentación que transmiten la información de la mejor manera posible.

Para ello, es necesario aumentar la capacidad de reflexión del sistema sobre su propio conocimiento de resolución de problemas, es decir, el modelo de sí mismo que tiene el sistema, con elementos propios de la interacción como son la evolución del diálogo y las características del usuario. Así, una misma pregunta puede ser respondida de formas distintas dependiendo del tipo de preguntas que la precedieron, del modo en que éstas fueron contestadas y de la clase de usuario que la ha formulado.

Entonces, esta tesis propone un modelo de interacción usuario-sistema que emerge como resultado de la interpretación de las necesidades de información del usuario en diferentes contextos y dominios y está soportado por una arquitectura cognitiva flexible. El modelo de razonamiento seguido para obtener las respuestas requeridas por el usuario se genera dinámicamente a



partir del planteamiento del problema o la pregunta formulada por el usuario así como de la disponibilidad de recursos cognitivos en el dominio en el que se plantea el problema.

## **5.2 Características del Modelo de Interacción Usuario-Sistema**

Alcanzar un nivel de comunicación efectivo entre un sistema y sus usuarios, requiere una etapa de especificación del diálogo en la que se busca la satisfacción de ciertos principios o características dirigidos a lograr el grado de cooperación entre ambos que requiere el dominio problema. En escenarios de interacción en los que el éxito en la consecución de un objetivo depende del grado de cooperación que pueda alcanzarse entre el sistema y el usuario, como son aquellos en los que el sistema constituye un soporte informativo de cara a la toma de decisiones por parte del usuario, estos principios adquieren especial relevancia.

### **5.2.1 Cualidades comunicativas del diálogo**

En este sentido, pueden enumerarse una serie de cualidades comunicativas deseables para los diálogos entre un sistema y sus usuarios que han de tenerse en cuenta a la hora de elaborar la especificación de dichos diálogos. Estas cualidades pueden clasificarse en tres grupos:

- aquellas relativas al aspecto cooperativo de la interacción y que son comunes a toda interacción comunicativa, ya sea ésta usuario-sistema o entre personas;
- aquellas específicas para la interacción de soporte a la toma de decisiones;
- aquellas referidas a aspectos propios de la usabilidad del sistema.

Con respecto a la primera categoría, [Grice, 75] enumera una serie de criterios a seguir para lograr una interacción cooperativa que están orientados a respetar el principio de cooperación, por el que el comportamiento comunicativo ha de

definirse en cada momento de acuerdo con los objetivos tácitos o expresos de la interacción que se esté llevando a cabo. Estos criterios son los siguientes:

- ♦ *Claridad.* La información a transmitir debe estar expresada de forma clara, ordenada y sin ambigüedades.
- ♦ *Relevancia.* La actitud adoptada en la comunicación debe ser relevante respecto a la dirección del intercambio, es decir, las acciones tomadas deben tener sentido dentro del contexto en el que se encuentre la interacción.
- ♦ *Credibilidad.* La información a transmitir ha de ser suficientemente verosímil.
- ♦ *Pertinencia.* La contribución al diálogo ha de ajustarse a las necesidades de la acción comunicativa que se lleve a cabo, es decir, no debe ser ni más ni menos informativa de lo necesario.

Los principios anteriores, aplicables para cualquier tipo de interacción comunicativa, pueden ampliarse y especializarse con otros más concretos relativos a un tipo de interacción específica, como es la interacción orientada a dar soporte en la toma de decisiones, descritos en [Jones, Mitchell, 95]:

- ♦ *Autoridad residente en el usuario.* El usuario debe ser el que articule el desarrollo del diálogo con el sistema y es el responsable último de las decisiones a tomar.
- ♦ *Honestidad.* La participación de un sistema en el proceso de toma de decisiones mediante su interacción con un usuario debe estar ajustada a sus capacidades, sin asumir compromisos para los que no está preparado, lo cual liga con el principio anterior ya que un sistema 'honesto' no pretenderá asumir responsabilidades que no le corresponden. Así pueden evitarse posibles confusiones en el usuario sobre el alcance de la ayuda que le puede proporcionar el sistema, haciendo posible que el usuario evalúe adecuadamente los riesgos de las posibles decisiones.



- ◆ *Transparencia.* El usuario, como responsable de las decisiones que se adopten, debería poder comprender no sólo las conclusiones ofrecidas por el sistema interlocutor sino también la forma en que ha llegado a ellas. Esto está en consonancia con el principio de honestidad anterior y la necesidad de que el sistema se muestre creíble y convincente respecto a sus conclusiones, para así generar confianza en el usuario y ofrecer un apoyo real en la toma de decisiones.
- ◆ *Inteligibilidad mutua.* La conducta comunicativa de un sistema debe estar orientada a garantizar la inteligibilidad mutua entre éste y el usuario, es decir, el usuario debe poder comprender qué es lo que está haciendo la aplicación y viceversa.
- ◆ *Recuperación de la inteligibilidad mutua.* Si en un momento de la interacción los objetivos o metas del usuario y el sistema dejan de estar en sintonía, entonces el sistema debería ser capaz de identificar el problema y recuperar el sentido de la interacción. Este tipo de problemas sólo suele presentarse en sistemas que hacen uso de lenguaje natural para comunicarse con el usuario, dado el carácter intrínsecamente ambiguo del mismo.
- ◆ *Múltiples perspectivas.* La aplicación debería ser capaz de ofrecer sus conclusiones desde múltiples perspectivas o con diferentes niveles de detalle del dominio de referencia de la interacción. Este requisito tiene su origen en la posibilidad de que haya diferentes usuarios que intervengan en la interacción con la aplicación, los cuales pueden que necesiten recibir la información desde diferentes puntos de vista.

Un tercer nivel de especificación de características deseables para el diálogo soportado por un sistema interactivo proviene del área de diseño de interfaces de usuario, en la que se considera como objetivo primordial del diseño del modelo de comunicación de un sistema el desarrollo de su *usabilidad* [Dix et al., 98]. Los principios que soportan la usabilidad se clasifican en tres categorías, en las que pueden identificarse principios más específicos:

⇒ *Facilidad de aprendizaje:* aspectos de los sistemas interactivos que facilitan a los nuevos usuarios el comprender cómo usarlos y cómo alcanzar un

máximo nivel de aprovechamiento o comunicación. Los principios que influyen en la facilidad para aprender un sistema son:

- ◆ *Predecibilidad*: cualidad de un sistema interactivo por la que el usuario puede determinar el efecto de una acción de interacción futura basándose en la historia de la interacción pasada. Este es un concepto centrado en el usuario que está asociado a la idea de comportamiento determinista del sistema desde la perspectiva del usuario. Entonces, la noción de predecibilidad puede asociarse con: (i) la habilidad del usuario para determinar el efecto de sus operaciones sobre el sistema, o (ii) la habilidad del usuario para saber qué operaciones pueden hacerse.
- ◆ *Capacidad de síntesis*: cualidad de un sistema interactivo por la que el usuario puede evaluar el efecto de las operaciones pasadas sobre el estado actual. Este concepto está relacionado con el anterior ya que la predecibilidad asume que el usuario tiene algún modelo mental de cómo se comporta el sistema, pero no dice nada de la forma en la que el usuario llega a formar ese modelo del comportamiento del sistema, lo cual requiere que el usuario pueda valorar las consecuencias de interacciones pasadas. Esto significa que cuando una operación cambia algún aspecto del estado interno del sistema, es importante que este cambio lo pueda percibir el usuario. El principio de honestidad se relaciona con la habilidad del interfaz de usuario para proporcionar un informe sobre tal cambio.
- ◆ *Familiaridad*: cualidad de un sistema interactivo que mide la correlación entre el conocimiento actual del usuario y el conocimiento requerido para llevar a cabo una interacción efectiva. Está relacionada con la impresión que el sistema produce en el usuario y cómo esta imagen da la pista al usuario sobre la forma de interactuar.
- ◆ *Capacidad de generalización*: cualidad de un sistema interactivo por la que el usuario puede aumentar su conocimiento sobre cómo interactuar con el sistema aplicando en una nueva situación el que ya tiene sobre interacciones específicas en situaciones similares. Este concepto puede verse como una forma de consistencia que facilita



también el que el usuario pueda hacerse con un modelo predictivo del sistema más completo.

- ♦ *Consistencia*: cualidad de un sistema interactivo que pone de manifiesto la semejanza en el comportamiento de la interacción en situaciones similares.

⇒ *Flexibilidad*: aspecto de los sistemas interactivos relativos a la multiplicidad de formas en las que un sistema y sus usuarios pueden intercambiar información. Los principios que contribuyen a la flexibilidad de la interacción con un sistema son:

- ♦ *Iniciativa de diálogo*: cualidad de un sistema interactivo por la que se libera al usuario de restricciones artificiales sobre el diálogo que impone el sistema para comunicarse con él. Si se entiende la interacción entre usuarios y sistemas como un diálogo, es importante considerar cuál de los dos interlocutores es el que tiene la iniciativa en el diálogo. Desde la perspectiva del usuario, un diálogo demasiado dirigido por el sistema disminuye la flexibilidad de la interacción, mientras que si lo conduce el usuario ésta se favorece.
- ♦ *Multiplicidad de líneas de diálogo*: cualidad de un sistema interactivo por la que se permite que la interacción usuario-sistema soporte más de una acción de interacción del usuario al mismo tiempo. Este concepto está relacionado con los diálogos multimodo, esto es, aquellos que utilizan más de un canal de comunicación para transmitir información.
- ♦ *Migración de tareas*: cualidad de un sistema interactivo por la que puede transferirse el control de la ejecución de una tarea entre el sistema y el usuario.
- ♦ *Multiplicidad de representaciones*: cualidad de un sistema interactivo por la que valores equivalentes admiten diferentes expresiones o representaciones tanto en la entrada como en la salida.
- ♦ *Capacidad de adaptación*: cualidad de un sistema interactivo por la que puede adaptar el interfaz de usuario basándose en la experiencia del

usuario o en la observación de la repetición de determinadas secuencias de acciones de interacción del usuario.

⇒ *Robustez*: aspecto de los sistemas interactivos relativo al nivel de soporte proporcionado al usuario para que éste alcance los objetivos previstos en el dominio de trabajo. Los principios que soportan la robustez de un sistema son:

- ◆ *Capacidad de observación*: cualidad de un sistema interactivo por la que el usuario puede evaluar el estado interno del sistema mediante la representación que éste hace en su interfaz.
- ◆ *Capacidad de recuperación*: cualidad de un sistema interactivo por la que el usuario puede tomar acciones correctivas para alcanzar sus objetivos una vez detectado un error en interacciones previas.

Principios generales de soporte de la interacción usuario-sistema		
Generales a cualquier interacción comunicativa		Específicas para la interacción soporte a la toma de decisiones
Claridad	Autoridad residente en el usuario	
Relevancia	Honestidad	
Credibilidad	Transparencia	
Pertinencia	Inteligibilidad mutua	
	Recuperación de la inteligibilidad mutua	
	Múltiples perspectivas	
Principios de soporte de la interacción usuario-sistema		
Asociados a la interacción comunicativa	Asociados a la interacción soporte a la toma de decisiones	Asociados a la interacción soporte a la toma de decisiones
Predecibilidad	Iniciativa de diálogo	Capacidad de observación
Capacidad de síntesis	Multiplicidad de líneas de diálogo	Capacidad de recuperación
Familiaridad	Migración de tareas	Capacidad de respuesta
Capacidad de generalización	Multiplicidad de representaciones	Capacidad de ajuste
Consistencia	Capacidad de adaptación	

Figura 5.2: Principios generales soporte de la interacción usuario-sistema



- ♦ *Capacidad de respuesta:* cualidad de un sistema interactivo que mide el nivel de comunicación entre el usuario y el sistema desde la perspectiva del usuario. El tiempo de respuesta se define en general como el tiempo que necesita el sistema para comunicar al usuario cambios de estado. Normalmente se desean tiempos reducidos y respuestas instantáneas, pero si esto último no es posible entonces el sistema debe ser capaz de notificar al usuario que ha recibido su solicitud y que está trabajando para generar una respuesta.
- ♦ *Capacidad de ajuste:* cualidad de un sistema interactivo que mide el grado en el que los servicios o funcionalidades del sistema soportan todas las tareas que el usuario desea llevar a cabo y en la forma en la que el usuario las entiende.

En el contexto de esta tesis, el conjunto de criterios considerados para configurar un modelo satisfactorio de interacción entre usuarios y sistemas resultan de la agregación de algunos de los principios anteriores relativos a la información a transmitir y de la selección y reinterpretación, desde la perspectiva de la operación, de los principios relativos al interfaz de usuario:

#### ♦ CONSCIENCIA

El rol que desempeñe un sistema en la interacción con un usuario debe ajustarse a sus capacidades, lo que implica que el sistema ha de hacerle ver al usuario cuál es el alcance de su ayuda. Así se evita confundir al usuario al hacer posible que éste evalúe adecuadamente los riesgos de sus decisiones, lo cual en ocasiones, como aquellas en las que se trabaja sobre modelos empíricos de la realidad o con información incompleta está reñido con respuestas únicas por parte del sistema.

Por ello, el sistema debe ser capaz de mostrarse creíble y convincente argumentando adecuadamente sus conclusiones y dejando al usuario la posibilidad de valorar si éstas son suficientes o no para apoyar sus decisiones y si necesita obtener información adicional.

### ◆ TRANSPARENCIA

Para facilitar que el usuario se construya un modelo mental del funcionamiento del sistema, además de poder recibir explicaciones sobre la forma en que el sistema ha alcanzado sus conclusiones debería poder tener una visión explícita del modelo de razonamiento empleado, es decir, debería tener acceso a una representación del modelo de comprensión del dominio que maneja el sistema.

Un sistema transparente, al facilitar que el usuario se construya un modelo mental del funcionamiento del sistema, hace posible también que éste pueda prever el efecto de sus interacciones con el sistema y comprender la relación entre interacciones pasadas y el estado actual, dando soporte a los principios de predecibilidad y capacidad de síntesis. Este concepto de transparencia está ligado también al principio de capacidad de observación, puesto que un aspecto fundamental de esa transparencia es dar acceso al usuario a una representación comprensible del estado de conocimiento del sistema.

### ◆ FAMILIARIDAD

Un aspecto importante para asegurar la usabilidad de un sistema, y que está relacionado con el principio de transparencia anterior, es la capacidad del sistema para expresar su conocimiento y sus conclusiones en un lenguaje próximo al que emplean sus usuarios. Este criterio da soporte también a los principios de multiplicidad de representaciones y capacidad de adaptación, por lo que el lenguaje con el que el sistema transmita su información debe determinarse en función de las características de los distintos usuarios.

### ◆ FLEXIBILIDAD

La evolución de un diálogo orientado a sustentar la toma de decisiones ha de ser lo suficientemente flexible como para considerar las consecuencias de razonar sobre planteamientos alternativos o de hacer rectificaciones a declaraciones manifestadas por el usuario con anterioridad. De esta manera podría obtenerse información adicional



que llevaría a reforzar los argumentos obtenidos a priori a favor o en contra de determinadas decisiones.

Este principio puede relacionarse con el principio de consciencia anterior ya que un sistema consciente o responsable debe serlo también con respecto a la participación del usuario. Parte de la ayuda para tomar las decisiones adecuadas pasa por reconocer y revisar interacciones previas o planteamientos erróneos del usuario que podrían llevar a error, haciendo que el modelo del estado de la interacción, asociado a una representación del estado de conocimiento alcanzado por el usuario y el sistema a lo largo del diálogo y que es común a ambos, se mantenga consistente y asegurando así la inteligibilidad mutua.

#### ◆ AGILIDAD

De forma similar a como sucede en las conversaciones entre personas, la calidad de la interacción entre un sistema y un usuario está relacionada con la eficiencia con la que se desarrolle esa interacción. No se puede hablar de interacción de calidad si ésta no se desarrolla con unos tiempos de respuesta razonables y de forma que las conclusiones alcanzadas durante el transcurso de la misma estén presentes o asumidas para ambos interlocutores.

#### ◆ VERSATILIDAD

La aplicación debería ser capaz de ofrecer sus conclusiones desde múltiples perspectivas o con diferentes niveles de detalle del dominio de referencia de la interacción. Este requisito tiene su origen en la posibilidad de que haya diferentes usuarios que intervengan en la interacción con la aplicación, los cuales puede que necesiten recibir la información desde diferentes puntos de vista.

Otro aspecto de este principio de versatilidad se refiere a la capacidad del sistema para adaptar su comportamiento a los distintos niveles de autoridad que pueden tener los usuarios. Se entiende que, dependiendo de las características del dominio, los usuarios pueden tener diferentes niveles de responsabilidad atendiendo a sus

diferentes capacitaciones, lo que implica que el tipo de información a la que pueden acceder y el nivel en el que pueden tomar decisiones varía. Entonces, el sistema debería ser capaz de gestionar el acceso a los mecanismos de toma de decisiones de cada uno de los tipos de usuarios en función de su nivel de autoridad.

Por otra parte, este principio convive con la idea general de que el usuario, en forma consistente con su nivel de autoridad, es el responsable último de las decisiones que se adopten en el dominio problema y por esta razón se entiende que a él le corresponde iniciar y dirigir el desarrollo del diálogo con el sistema aunque en ocasiones podría decidir traspasar temporalmente esa iniciativa al sistema. Por ejemplo, si la aplicación está orientada a monitorizar el funcionamiento de un sistema con varios componentes, el usuario podría centrar su atención en uno de ellos dejando al sistema encargado de notificarle la aparición de eventos inesperados o situaciones problemáticas en los otros componentes. Este aspecto está ligado además con el principio de multiplicidad de líneas de diálogo, facilitada por el uso generalizado de interfaces de usuario multimedia, ya que por ejemplo la interacción dirigida a supervisar en detalle uno de esos componentes podría llevarse a cabo en lenguaje natural mientras que la comunicación del estado de los demás componentes podría realizarse mediante códigos de colores, sonidos o mensajes de alerta.

Aunque el objetivo al depositar fundamentalmente la iniciativa de diálogo en el usuario es favorecer la flexibilidad de la interacción, es necesario encontrar un equilibrio de manera que se adecue esta flexibilidad a las características del usuario. Así, podría entenderse que un usuario inexperto necesitase de una mayor participación del sistema en el desarrollo del diálogo, mientras que un usuario experimentado conoce bien las funcionalidades del sistema y por tanto puede dirigir la interacción directamente hacia sus objetivos.

Este concepto de versatilidad incorpora los principios de inteligibilidad mutua, multiplicidad de representaciones y capacidad de adaptación.



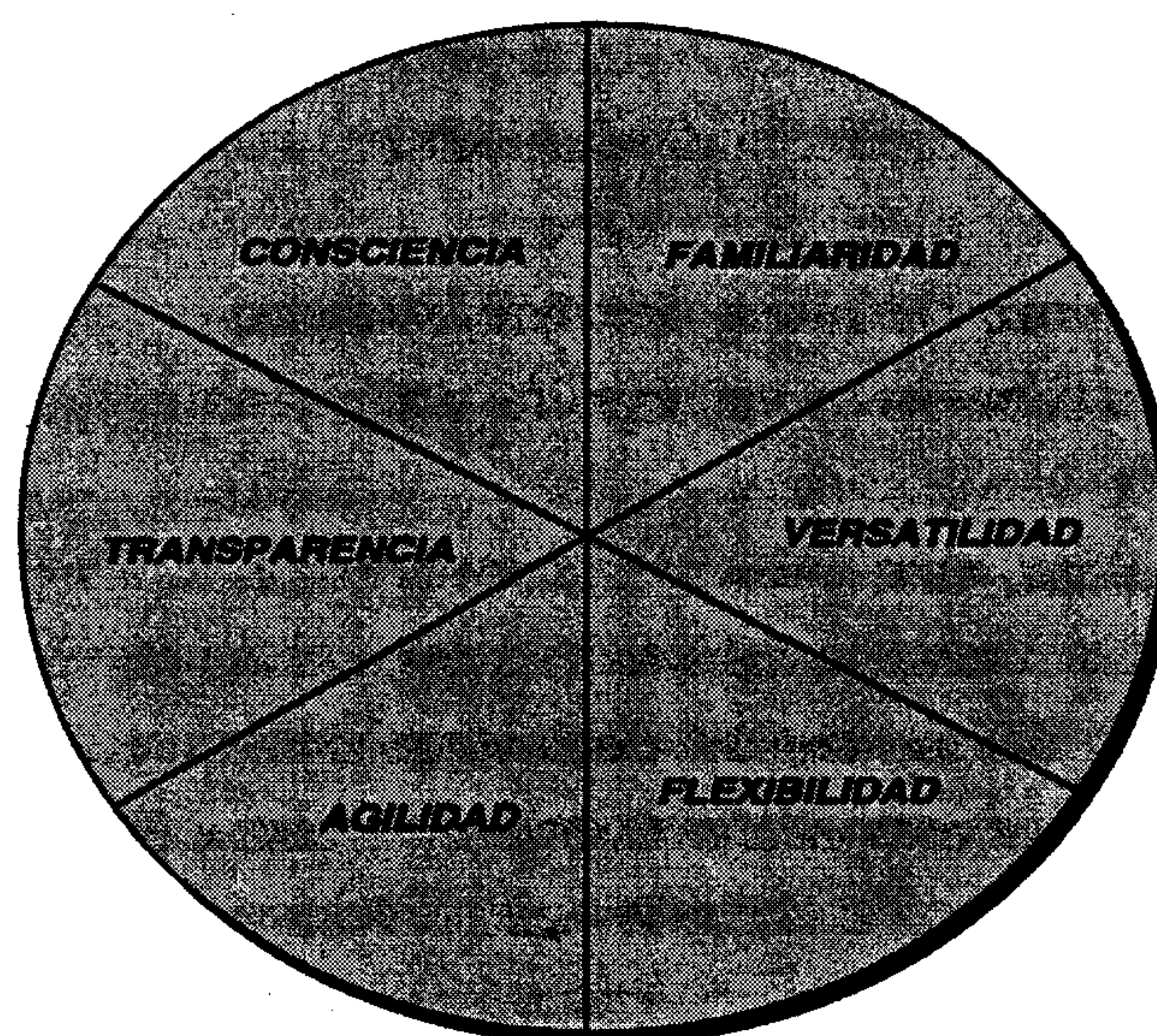


Figura 5.3: Principios de la interacción usuario-sistema avanzada

### 5.2.2 Comportamiento del sistema

La satisfacción de los principios anteriores por parte de un sistema de ayuda a la decisión requiere de éste determinados comportamientos que pueden caracterizarse como sigue:

- Al asumir la autoridad del usuario y su responsabilidad sobre las decisiones que se adopten, el rol del sistema en su interacción con el usuario queda circunscrito al de un *asistente*, que ofrece la información necesaria para tomar las decisiones más adecuadas pero que no reemplaza al usuario en la elección de las mismas.
- En relación con el anterior, en el caso en el que un mismo sistema dé servicio a varios tipos de usuarios, con diferentes niveles de responsabilidad, sería deseable que el sistema pudiese graduar su nivel de asistencia a las características de sus diferentes interlocutores, adaptando su lenguaje y la expresión de sus conclusiones a las características de los diferentes usuarios con el objetivo de satisfacer los principios de transparencia, familiaridad y versatilidad.
- El principio de consciencia puede entenderse satisfecho si el sistema tiene capacidad de evaluar su propio funcionamiento, esto es, si es capaz de acompañar sus conclusiones con justificaciones bien



argumentadas de su validez o con alguna medida o valoración de la confianza que tiene en sus respuestas en función del contexto de conocimiento en el que opera, pudiendo ofrecer en su caso alternativas al usuario si dicha respuesta no es única o si no tiene una confianza total sobre la misma.

- Para conseguir desarrollar un diálogo usuario-sistema flexible es necesario que el sistema sea capaz de dar respuesta a preguntas que no necesariamente siguen la línea argumental iniciada con el diálogo sino que se plantean bajo diferentes hipótesis, dando lugar en algunos casos a contradicciones con el estado de creencias alcanzado durante el diálogo que el sistema debería poder detectar y comunicar al usuario para reconducir la interacción.
- Para satisfacer el principio de versatilidad y conseguir agilidad en el desarrollo de la interacción usuario-sistema, el sistema debería saber interpretar qué tipo de usuario es su interlocutor y, a la vista de la evolución reciente de la interacción, cuáles son las intenciones que persigue con el diálogo, para así adaptar su comportamiento a los objetivos de éste y ofrecer sus respuestas en el modo más adecuado para las características del mismo.

A la vista de las características deseables en los diálogos entre sistemas de ayuda a la decisión y sus usuarios, y el comportamiento que han de mostrar estos sistemas para ofrecer dichas cualidades, es necesario identificar qué deben saber estos sistemas para alcanzar los objetivos de interacción propuestos. Una vez identificado este conocimiento queda por ver cómo se distribuiría entre diferentes componentes de un sistema y, cómo se estructurarían y combinarían dichos componentes para alcanzar el modelo de comunicación descrito.





## 6. Modelo de Conocimiento Para Soportar Interfaces Inteligentes

---

En este capítulo se presenta el modelo de interacción usuario-sistema que propone esta tesis para mejorar la capacidad comunicativa de los denominados sistemas inteligentes en contextos de operación de toma de decisiones. A la vista de las metas marcadas en el capítulo anterior respecto a la interacción usuario-sistema, en este capítulo se definen los requisitos del modelo que ha de servir para alcanzar esos objetivos y se plantea una arquitectura, llamada ROHCI (*Reasoning Oriented to Human-Computer Interaction*), que satisface dichos requisitos, soportando el tipo de comunicación deseado. Después, se explican con detalle cada uno de los componentes de dicha arquitectura.

### 6.1 Arquitectura General

Un diálogo entre un usuario y un sistema de ayuda a la decisión es una sucesión de preguntas y respuestas - salvo excepciones, planteadas por el usuario y respondidas por el sistema - orientadas a proporcionar al usuario la suficiente información para facilitarle las operaciones que haya de llevar a cabo en el dominio problema, en función de sus intereses y responsabilidades; lo cual implica conocer a los distintos tipos de usuarios y ante una misma pregunta, adaptar la respuesta a esos intereses y responsabilidades.

En su concepción más básica, la posibilidad de ofrecer diferentes respuestas a una misma pregunta puede estar soportada o bien (i) por un módulo generador de presentaciones inteligentes que a una misma información a transmitir le da la forma más adecuada para el tipo de usuario que formula la pregunta, o bien (ii) por un conjunto de métodos de resolución de problemas, donde cada uno de ellos es generador de la información relativa a diferentes



clases de respuestas que posteriormente podrían ser presentadas al usuario por un interfaz convencional. En una versión más avanzada, la combinación de ambas posibilidades daría lugar a un sistema con una capacidad comunicativa muy superior a la de los sistemas disponibles actualmente al integrar el *know-how* del sistema y el modelo de las necesidades del usuario.

La línea de trabajo de esta tesis camina en esta dirección, considerando que la interacción entre un sistema y sus usuarios es el producto de un diálogo entre ambos, centrándose en la generación de diferentes respuestas a partir de un conjunto de mecanismos resolvedores de problemas distintos, y poniendo a disposición del generador de presentaciones tanto las diferentes respuestas asociadas a una misma pregunta junto con sus explicaciones como los criterios de interacción que se tuvieron en cuenta para generar esa información concreta, que podrían ser utilizados por el interfaz para determinar la forma adecuada de presentarla.

La obtención de una respuesta supone resolver un problema, normalmente siguiendo una serie de pasos que definen a su vez subproblemas más sencillos, cuya solución requiere seleccionar en cada uno de los pasos cuál de los posibles métodos potencialmente aplicables es el más adecuado. Por ello es necesario disponer de una inteligencia capaz de hacer esta selección de forma dinámica en función del tipo de usuario y el estado de la interacción. El tipo de razonamiento que realiza esta inteligencia se apoya por tanto en conocimiento sobre conocimiento de resolución de problemas, y por ello diremos que se trata de un *razonador metanivel*, pero también en hipótesis sobre las características del diálogo. Entonces, el razonador metanivel tendrá conocimiento de dos tipos:

- De *caracterización*, para identificar el tipo de diálogo que mantiene con un usuario y así generar la respuesta en forma consistente con el modo de interactuar observado;
- De *decisión*, para seleccionar el método que ha de aplicarse para cada problema a partir de requisitos del diálogo y de la disponibilidad de conocimiento del dominio.

Este aspecto es una particularidad del razonamiento metanivel que plantea esta tesis ya que en los modelos existentes que incorporan algún mecanismo de selección dinámica de formas de resolver problemas, los criterios empleados

para decidir el método adecuado están siempre basados en aspectos de disponibilidad o características de conocimiento del dominio y en algún caso en la sintonía entre el método a elegir y métodos aplicados con anterioridad.

Por otra parte, si se desea alcanzar un diálogo más realista entre usuarios y sistemas es necesario capacitar al sistema para revisar el conjunto de sus creencias a medida que éste va avanzando. Si se analiza la evolución de una conversación entre personas, puede observarse que la incorporación de nueva información deducida o justificada por ciertas hipótesis hace que el conjunto de creencias de los interlocutores varíe. Esta variación puede suponer aumentar dicho conjunto o bien puede que implique revisar, es decir eliminar, parte de esas creencias como hechos ciertos, al identificar una incompatibilidad entre las nuevas hipótesis e hipótesis asumidas en momentos anteriores del diálogo. Por tanto, hay que dotar al sistema de un mecanismo con el que mantener consistente su estado de información de manera que por una parte se agiliza el diálogo al mejorar la eficiencia del proceso de resolución de problemas al poder tomar información de dicha memoria sin necesidad de volver a generarla, y por otra se aumenta la flexibilidad del diálogo al incorporar la no monotonía en la gestión de la memoria.

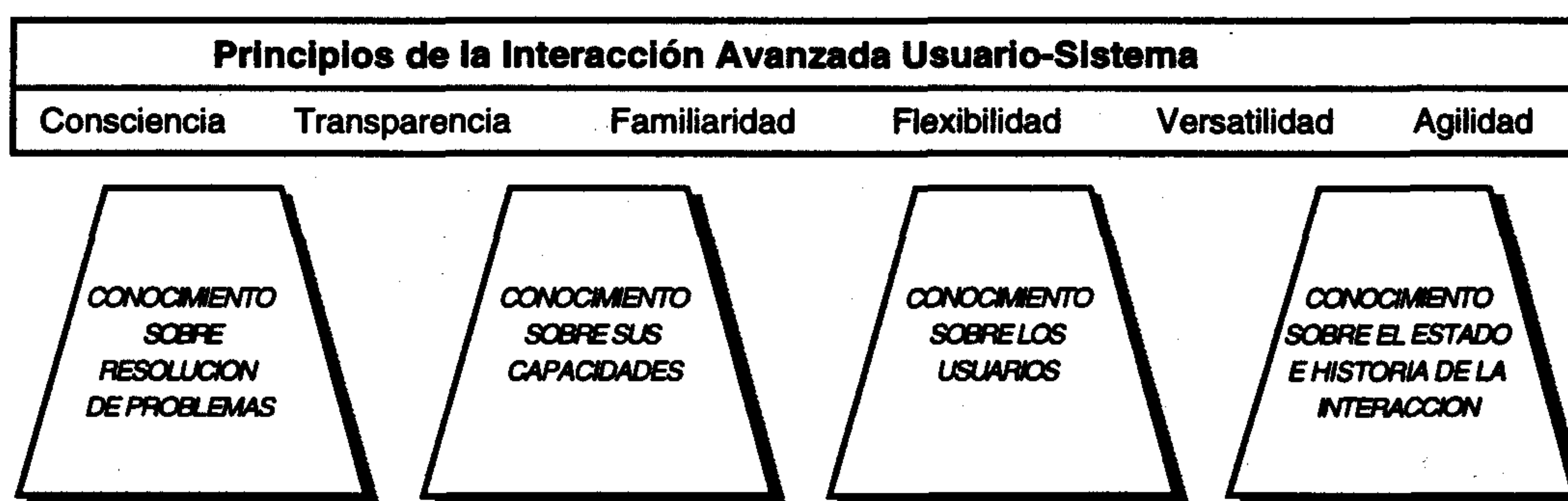
En resumen, el tipo de interacción que se persigue entre un sistema y sus usuarios supone dotar al sistema de las siguientes clases de conocimiento:

- ⇒ El sistema debería saber cómo resolver los distintos problemas que se le pueden presentar en el dominio en el que se toman las decisiones, es decir, qué mecanismos puede aplicar para encontrar soluciones. Cuanto más rico sea este conocimiento mayor variedad de respuestas podrán ofrecerse y mayor será la posibilidad de enriquecer la interacción al poder elegir el modo de responder a los usuarios. Además, la formulación de este conocimiento de resolución de problemas debería ser tal que le permitiese al sistema elaborar explicaciones sobre la forma en que alcanza sus conclusiones, por lo que aunque se favoreciese el uso de métodos heurísticos basados en conocimiento declarativo no deberían descartarse otro tipo de métodos, matemáticos por ejemplo, si fuesen relevantes en el dominio de aplicación.
- ⇒ El sistema debería saber con qué clase de usuarios puede comunicarse, qué tipo de operación podrían llevar a cabo estos usuarios, es decir, qué



clase de objetivos podrían estar persiguiendo cada uno de ellos cuando entablan un diálogo con el sistema, y aspectos relativos a la forma en que desean mantener ese diálogo. Haciendo uso de este conocimiento podría elaborar y adaptar sus respuestas en función de las características y objetivos del interlocutor, consiguiendo así agilizar el diálogo al evitar que los usuarios tengan que realizar consultas adicionales para conseguir la información adecuada a sus necesidades y características.

- ⇒ El sistema debería saber sobre sus propias capacidades de resolución de problemas, es decir, en qué forma puede utilizar lo que sabe sobre cómo resolver problemas para satisfacer las demandas de un tipo de usuario concreto; lo que implica conocer las peculiaridades de los distintos métodos de resolución de problemas de que dispone y las circunstancias en las que son adecuados para generar respuestas. Este conocimiento hace posible flexibilizar enormemente el diálogo al dar la posibilidad de ofrecer diferentes respuestas a una misma pregunta.
- ⇒ El sistema debería saber cuál es su estado de creencias en cada momento de su interacción con un usuario - cuáles son las conclusiones que se han alcanzado a partir de información proporcionada por el usuario o por otro sistema, lo que constituye una memoria de la interacción. Esto implica saber cómo ampliar esta memoria, según transcurre el diálogo, de manera que el nuevo conocimiento sea consistente con el que ya se poseía, y en caso contrario saber dónde reside la inconsistencia y cómo resolverla, por lo que ha de poseer también conocimiento sobre incompatibilidades entre creencias del sistema respecto al dominio problema. De esta forma se contribuye también a flexibilizar y agilizar el diálogo al disponer de un depósito de información que puede recuperarse si se solicita sin necesidad de volver a generarla.



*Figura 6.1: Áreas de conocimiento soporte de los principios de interacción usuario-sistema*

Entonces, la definición de una arquitectura que soporte el modelo de interacción deseado manejando los distintos cuerpos de conocimiento anteriores, representados en la figura 6.1, podría plantearse a partir de los siguientes elementos:

- una colección de métodos de resolución de problemas soportados por conocimiento del dominio y organizados como modelos estructurados de conocimiento;
- un sistema reflexivo de resolución de problemas a metanivel que dispone de (i) un modelo de los escenarios de interacción del sistema con los diferentes usuarios y de (ii) un modelo de las características y propiedades del conjunto de métodos de resolución de problemas. Ambos modelos se utilizarían para diseñar dinámicamente la estructura de inferencia que ha de ejecutarse para obtener las respuestas a partir de las conclusiones extraídas del modelo de escenarios de interacción y de las características del diálogo.
- un sistema de gestión de la memoria de la interacción responsable de mantener consistente el conjunto de creencias del sistema.

Esta arquitectura se denomina ROHCI (*Reasoning Oriented to Human-Computer Interaction*) y está representada en la figura 6.2.

En líneas generales, la operación conjunta de los componentes anteriores sería la siguiente:

- (1) La pregunta planteada por el usuario se analiza para obtener: (i) la clase de problema que hay que resolver para ofrecer una respuesta, (ii) el conjunto de hipótesis sobre el que se plantea la consulta y, (iii) requisitos iniciales sobre el modo de interacción que el usuario desea tener con el sistema, es decir, indicaciones sobre la forma en la que el usuario desea que se le responda.
- (2) El gestor de la memoria de la interacción analiza el conjunto de hipótesis planteadas en la pregunta para detectar posibles incompatibilidades con otras hipótesis ya consideradas en etapas



previas del diálogo. Si éstas aparecen se le notifica al usuario y se actúa en consecuencia, replanteando la pregunta.

- (3) El razonador metanivel a la vista de la clase de pregunta formulada, la historia reciente del diálogo y el modo de interacción deseado por el usuario selecciona un conjunto completo de requisitos de interacción, consistentes con las características del usuario, a tener en cuenta en la generación de la respuesta. Dichos requisitos, junto con la información disponible en la memoria y el conocimiento sobre las propiedades de los resolvers de problemas disponibles, se utilizan para diseñar el modelo de razonamiento que satisface el mayor número de ellos.

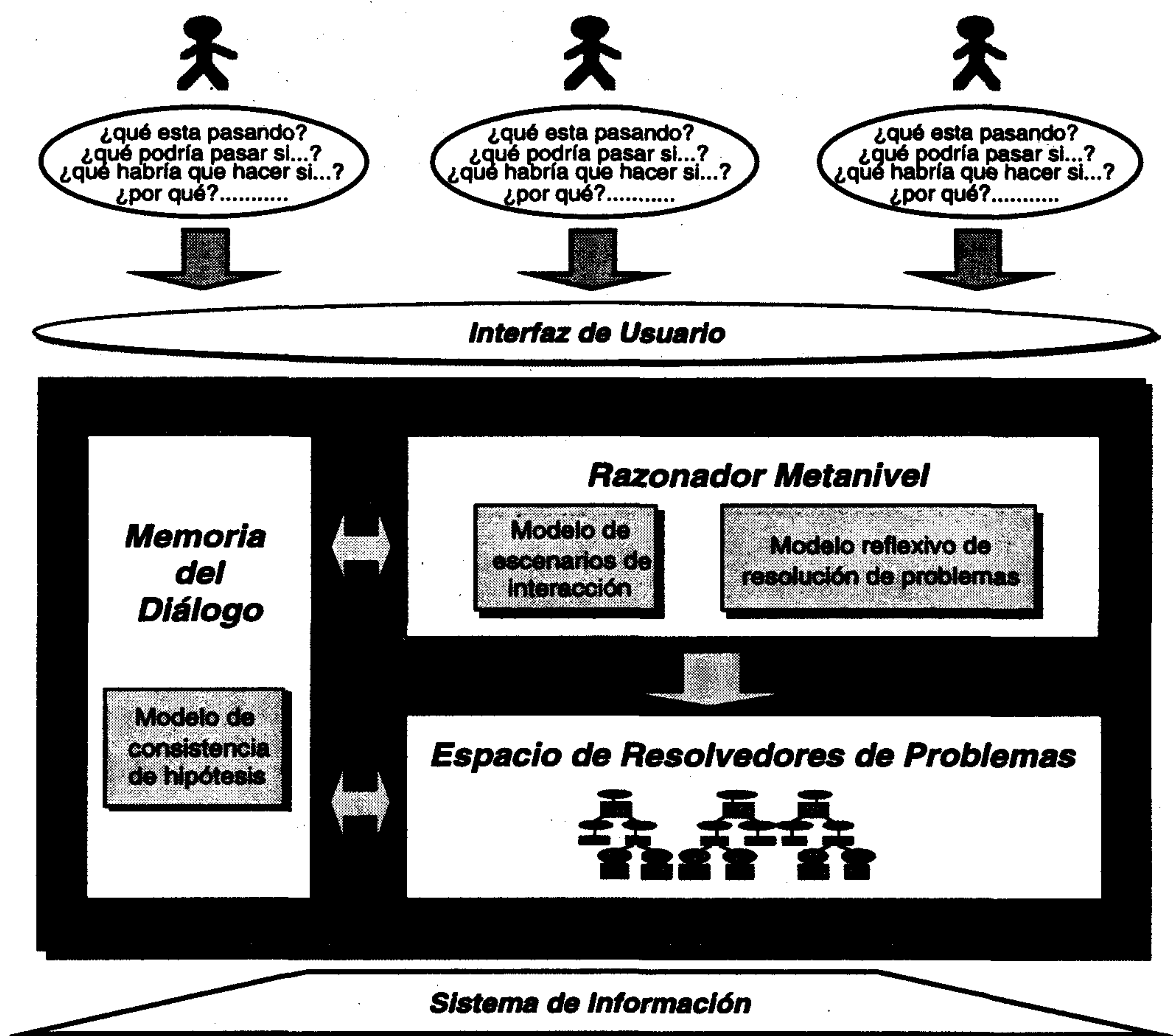


Figura 6.2: Arquitectura general ROHCI

- (4) El modelo de razonamiento definido anteriormente se aplica ejecutando en el orden preciso los resolvers de problemas seleccionados hasta generar la respuesta buscada.
- (5) La información generada al ejecutar el modelo de razonamiento se le pasa al gestor de la memoria para que la incorpore al estado de creencias del sistema y al interfaz de usuario para que elabore la presentación más adecuada.

En las siguientes secciones se hace una descripción detallada de cada uno de los elementos anteriores.

## 6.2 El Modelo de Interacción

Al entender que la interacción entre un sistema y sus usuarios se establece en torno a un diálogo, se hace necesario especificar en primer lugar los términos en los que puede desarrollarse ese diálogo para posteriormente poner los medios con los que articularlo. Es por ello necesario identificar y clasificar los diferentes tipos de preguntas que el usuario puede plantear al sistema mediante las cuales satisfacer sus necesidades de información en el dominio problema. La especificación de estas preguntas va acompañada de la caracterización de la correspondiente respuesta; es más, podría entenderse que la concepción de la respuesta es previa a la pregunta, al especificar ésta los requisitos de información del usuario y plantearse la pregunta como un medio para solicitar esa información.

La definición del modelo de interacción usuario-sistema se lleva a cabo en varias etapas:

- (1) Un primer paso consiste en identificar las clases de preguntas relevantes en función de los objetivos del sistema o del problema al que se desee dar solución. Por ejemplo, si la aplicación se diseña con el objetivo de diagnosticar el comportamiento de un sistema físico cabe considerar dos clases de preguntas: la clase *qué está pasando?* mediante la que se accede a información sobre el estado actual del sistema y sus componentes, y la clase *qué podría pasar?* con la que pueden obtenerse



datos sobre el estado previsible a corto plazo de este sistema y sus componentes.

- (2) Después, dentro de cada una de estas clases pueden incluirse varias preguntas de diferentes tipos, por ejemplo, preguntas de la clase *qué está pasando?* podrían ser *qué está pasando en el sistema?* y *qué está pasando en <componentes>?*, donde esta segunda podría dar lugar a varias preguntas distintas al considerar uno o más de los diferentes componentes del sistema. Asimismo, se consideran dentro de esta clase todas las preguntas del tipo *por qué?* asociadas a las preguntas anteriores mediante las que se busca obtener una explicación de las respuestas ofrecidas para esas preguntas. En el caso de la clase *qué podría pasar?*, además de las preguntas *qué podría pasar en el sistema?* y *qué podría pasar en <componentes>?*, podrían incluirse preguntas condicionales en las que se consideran hipótesis sobre estados del entorno en el que funciona el sistema como *qué podría pasar en <componentes> si sucede <condiciones externas>?*.
- (3) Por último, queda expresar el problema asociado a la búsqueda de las respuestas en el lenguaje propio de la resolución de problemas, es decir, para cada una de las preguntas hay que definir cuál es la tarea que debe resolverse para encontrar la solución, por ejemplo, para encontrar la respuesta a una pregunta del tipo *qué está pasando en <componentes>?* podría tener que resolverse la tarea *clasificar\_estado* a la que se le pasa como argumento el componente o componentes sobre los que ha de identificarse el estado.

Aunque la expresión ideal del conjunto de preguntas de este modelo de interacción sería mediante el uso de lenguaje natural, la complejidad de esta solución y los objetivos perseguidos en esta tesis, que están más orientados a mejorar los aspectos de contenido de la interacción más que los de forma, han conducido a la aplicación de un sistema de menús estructurado como mecanismo mediante el que facilitar al usuario la formulación de las diferentes preguntas del modelo.

En función del punto en el que se encuentre un diálogo y de la información sobre la estructura de los posibles diálogos usuario-sistema de que disponga el interfaz de usuario, el sistema le muestra al usuario los diferentes tipos de

preguntas que puede plantear en ese contexto del diálogo y entre éstas el usuario puede elegir una seleccionándola con el ratón.

Dado que puede haber preguntas con variables, como *qué podría pasar en <componentes>?*, la especificación de la pregunta concreta se logra en este caso mediante otros menús que se despliegan al pasar el ratón sobre la superficie de la variable, por ejemplo, al pasar el ratón sobre <componentes> se desplegaría un menú con todos los componentes del sistema sobre los que puede solicitarse información de su estado futuro.

## 6.3 El Espacio de Resolvedores de Problemas

### 6.3.1 Elementos

Una consulta de un usuario le plantea a un sistema una tarea a resolver: encontrar el modo de obtener la información solicitada. El procedimiento a seguir para obtener la respuesta será un método de resolución de problemas, que puede ser simple y llevar a cabo un proceso de inferencia básico que use conocimiento del dominio o bien puede requerir resolver varios subproblemas más sencillos para los que a su vez deberían existir métodos que puedan ofrecer soluciones. Por tanto, la generación de una respuesta pasa por la elección del método que llevará a cabo la tarea asociada a la pregunta.

En los sistemas tradicionales que hacen uso de estos conceptos de tarea y método, esta elección suele ser inmediata ya que la relación entre tareas y métodos es unívoca. Sin embargo, cabe la posibilidad de que una misma tarea pueda ser realizada por diferentes métodos, por lo que han de introducirse criterios en los que apoyarse para seleccionar el método más adecuado. Por ejemplo, si un usuario pregunta por el estado del servicio en una línea de autobuses, el sistema sabe que tiene que caracterizar la situación de la línea, para lo cual podría aplicar un método que primero analizara datos recogidos en la línea sobre el estado de los vehículos o del tráfico, y después clasificara la situación de la línea en virtud de los resultados del análisis. Sin embargo, también podría emplear otro método algo más robusto que antes de analizar los datos los estudiase para ver si están completos y son fiables, y si no lo son completarlos o en su caso recalcularlos.



Por tanto, ante la necesidad de responder a una consulta de un usuario se plantea la dificultad de determinar la forma en que debe generarse la respuesta. Si un número significativo de las tareas que deben realizarse para obtener las respuestas en un dominio problema pueden llevarse a cabo con métodos alternativos, el conjunto de métodos de resolución de problemas asociados a la aplicación adquiere una dimensión y complejidad significativa, que podríamos denominar espacial, en contraposición a la visión plana de los modelos en los que cada tarea se lleva a cabo con un único método. Es por ello que hablaremos de ese conjunto de métodos como de un *Espacio de Resolvedores de Problemas (ERP)*.

El hecho de disponer de varios métodos para generar una misma respuesta implica necesariamente que habrá redundancia en el conocimiento sobre el dominio problema. Aunque por una parte, mantener la consistencia de este conocimiento tiene un coste que ha de ser considerado, por otra refleja una realidad en dominios problema complejos en los que ante la diversidad de situaciones conflictivas se aplican diferentes formas de resolver problemas adecuadas a la clase de situación vigente. Además, con los avances experimentados en la investigación de la reusabilidad de componentes software, el coste en la realización de un sistema con múltiples métodos podría verse reducido considerablemente si el soporte computacional se obtuviese en buena medida de métodos de resolución de problemas predefinidos extraídos de una librería de métodos.

### 6.3.2 Estructuración

Dado que se va a trabajar con una colección de métodos de resolución de problemas para generar las respuestas, se hace necesario aplicar un modelo de organización de estas formas de resolver orientado a proporcionar dichas respuestas en la forma que requiere el diálogo. A la hora de plantearse la organización de estos elementos hay que tener en cuenta dos aspectos fundamentales:

- requisitos globales que debe verificar el modelo de organización desde el punto de vista de calidad de producto informático;
- diversidad del conocimiento que ha de ser estructurado, tanto en su naturaleza como en su formulación y representación.

Como requisitos globales podrían enumerarse los siguientes:

- ⇒ que pueda soportar un proceso de razonamiento eficaz de cara a la operación en tiempo real, es decir, que permita procesar eficientemente los métodos que deben ejecutarse;
- ⇒ que tenga una arquitectura flexible que facilite las tareas de mantenimiento, minimizando el coste de suprimir, modificar o añadir elementos en la organización, y
- ⇒ que sea capaz de proveer explicaciones que justifiquen sus conclusiones apoyándose en la estrategia de razonamiento seguida. Este aspecto es fundamental para conseguir mejorar la calidad de la comunicación entre el sistema y el usuario.

Por otra parte, ha de tenerse en cuenta que en dominios complejos puede ser necesario realizar diferentes tipos de razonamiento, como clasificar, diagnosticar, predecir, diseñar, etc. Estas variadas formas de razonar estarán soportadas por diferentes tipos de conocimiento, especializado en áreas específicas del dominio y formulado a distintos niveles de abstracción, como pueden ser bases de conocimiento que contienen información de detalle en contraposición a otras que incluyen conocimiento estratégico. En este sentido, cabe destacar una división clara entre el conocimiento que expresa lo que se sabe de un dominio, el cual es de tipo declarativo y estará contenido en bases de conocimiento, y el conocimiento que determina cómo razonar con ese conocimiento sobre el dominio, que es de tipo procedimental y está asociado al control sobre la secuencia de ejecución de los pasos de un proceso de razonamiento. Además, al plantearse un modelo de interacción usuario-sistema capaz de responder una misma pregunta de formas distintas se está introduciendo necesariamente redundancia en el conocimiento, por lo que la complejidad del modelo aumenta, lo cual hace aún más necesario la satisfacción de los requisitos globales anteriores.

Es por tanto necesario proporcionar visiones explícitas de este conocimiento al usuario a diferentes niveles de abstracción, con objeto de facilitar la comunicación usuario-sistema, lo que enlaza con la visión abierta de los sistemas basados en el conocimiento (SBC), que son inspeccionables y capaces de dar explicaciones.



La solución adoptada para poder llevar a cabo lo anterior se apoya en las ideas sobre estructuración del conocimiento procedentes de la comunidad de adquisición del conocimiento, y en particular en las estructuras de tareas genéricas [Chandrasekaran et al., 92] (ver apartado 2.1.2). El formalismo empleado se basa en las ideas de tarea, método y subtarea, haciendo una distinción explícita entre tarea y método para posibilitar la asociación de múltiples métodos a una misma tarea. Según esta aproximación, una tarea define un problema a resolver, es decir, la obtención de determinados resultados a partir de un conjunto de premisas. Un método, como ya se ha mencionado anteriormente, define un procedimiento con el que resolver una tarea. Si este procedimiento requiere alcanzar diversos subobjetivos, cada uno de ellos definirá un subtarea que a su vez podrá ser llevada a cabo por uno o más métodos. Por el contrario, si el procedimiento que define el método no puede descomponerse en distintos pasos, se dice que realiza un inferencia básica soportada directamente por el conocimiento del dominio.

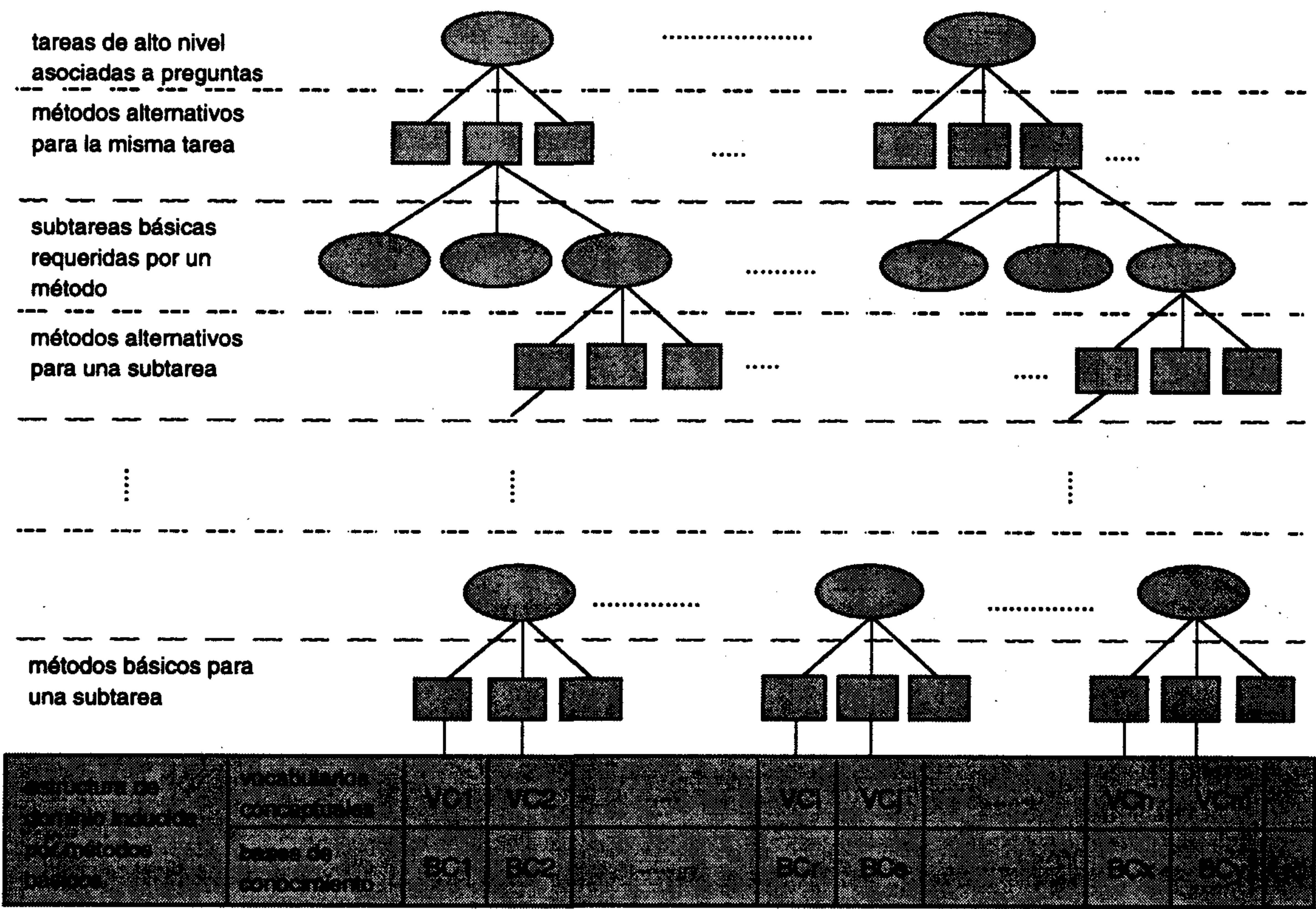


Figura 6.3: Estructuración del conocimiento en el Espacio de Resolvedores de Problemas

La aplicación de este formalismo al ERP define una colección de estructuras jerárquicas de conocimiento, construidas a partir de las tareas de más alto nivel con un proceso de descomposición jerárquica tarea-método-subtarea, y que en su nivel inferior incluyen los métodos básicos junto con el conocimiento que les da soporte (ver figura 6.3). Con este modelo de estructuración del conocimiento se ofrece, por una parte, una visión a alto nivel de las posibles estrategias de razonamiento que el sistema puede realizar para resolver problemas, y por otra, se organiza el conocimiento del dominio de forma altamente modular haciendo posible preservar sus diferentes características y al mismo tiempo ofrecer una perspectiva del proceso o procesos de razonamiento en los que participa.

De esta manera, el nivel de interacción usuario-sistema puede superar al de los sistemas convencionales al permitir al usuario inspeccionar el conocimiento del sistema y la forma en que éste lo emplea para resolver problemas, y ofrecer al sistema la posibilidad de generar explicaciones sobre sus conclusiones, justificadas por la estrategia de razonamiento seguida y el conocimiento del dominio empleado.

Esta organización modular no es simplemente una forma de estructurar conocimiento sino que aporta además una visión de orden superior que permite expresar, utilizando un medio descriptivo natural e intuitivo, lo que se sabe sobre el propio conocimiento lo cual proporciona una base muy adecuada sobre la cual apoyar la visión reflexiva que requiere el proceso de comunicación que se plantea en esta tesis.

## **6.4 El Razonador Metanivel**

Los aspectos de adaptabilidad al usuario, flexibilidad y agilidad del diálogo, relevantes para conseguir un buen nivel de interacción usuario-sistema, plantean la necesidad de dotar al sistema de la capacidad de reflexionar sobre la forma en que debe participar en el diálogo con un usuario. Esto supone proporcionarle conocimiento sobre sus capacidades y limitaciones en la resolución de problemas, así como el modo de aprovechar dichas capacidades para satisfacer las demandas de un usuario en un contexto determinado de un diálogo. Entonces, un elemento clave de la arquitectura propuesta es el módulo



de razonamiento reflexivo que posee este conocimiento y sabe cómo utilizarlo para alcanzar los objetivos comunicativos planteados durante la interacción. Las secciones siguientes describen este elemento en detalle.

### 6.4.1 Descripción general

Decidir cuál es el tipo de respuesta que debe darse a un usuario, de entre todas las posibles, implica decidir cómo resolver el problema de generar la información que va a ofrecerse como respuesta, de entre todas las formas conocidas de resolverlo. Además, como cada una de las diversas maneras de resolver un problema puede necesitar llevarse a cabo en una serie de pasos que a su vez constituyen subproblemas, el sistema puede encontrarse ante varios puntos de decisión antes de llegar a configurar el modo de alcanzar la respuesta. La formulación del conocimiento mediante el que expresar la metodología a seguir para resolver un problema, y en definitiva estas decisiones, hace uso necesariamente del vocabulario con el que se identifican los problemas y las formas de resolverlos, es decir, ese vocabulario incluirá conceptos relativos a tareas y métodos.

Entonces, como se mencionó al final del apartado anterior, el hecho de organizar el conocimiento del ERP como una estructura de tareas y métodos, además de las ventajas propias de dicha organización (modularidad, facilidad de mantenimiento, etc.), ofrece la ventaja añadida de proporcionar parte del lenguaje con el que expresar el conocimiento reflexivo. La otra parte del lenguaje es la que se utiliza para expresar los criterios en los que basa el sistema su elección de uno u otro método para resolver un problema, es decir, una tarea.

Como se mostró en el capítulo 2, las propuestas más recientes sobre criterios de selección de métodos están basadas en hipótesis sobre las capacidades de los métodos y las características del dominio y el entorno en el que se apliquen. Sin embargo, la evaluación de estas hipótesis se hace durante la etapa de diseño de la aplicación, quedando el modelo de razonamiento del sistema final completamente determinado, con cada tarea asociada a un único método. Por otra parte, las soluciones de selección de métodos en tiempo de ejecución, presentadas también en ese capítulo, están basadas en el análisis del contexto en el que ha de ejecutarse el método, observando si el estado de resolución de problemas del sistema dispone del conocimiento que requiere el método y en

algunos casos considerando la sintonía de los métodos candidatos con otros aplicados previamente.

En esta tesis se plantea un modelo reflexivo de selección dinámica de métodos que basa sus decisiones en hipótesis, no sólo sobre el contexto y la historia de la resolución de problemas, sino fundamentalmente sobre características del diálogo mantenido con el usuario. Dicho modelo ha sido aplicado y validado sobre un ejemplo real, y se presenta en los capítulos 7 y 8. El hecho de decidir la forma de responder, o lo que es lo mismo de resolver un problema, a partir del análisis de la forma de dialogar es un aspecto original y constituye el soporte de lo que se considera un modelo de interacción enriquecido, que permite proporcionar una adecuada flexibilidad y agilidad en un contexto de toma de decisiones.

El razonador metanivel que construye la estructura de razonamiento con la que generar una respuesta debe pasar por las siguientes etapas:

- (1) una vez identificada la tarea genérica asociada a la pregunta formulada por el usuario, el objetivo es analizar las características del usuario actual y del diálogo reciente para determinar las máximas comunicativas que deben guiar la selección de los métodos,
- (2) diseñar de forma dinámica el modelo de razonamiento que debe ejecutarse a partir de la tarea y el modo de interacción identificados previamente.

El primer paso es sencillo ya que se considera que cada clase de preguntas está asociada a una única tarea, por tanto la identificación de la tarea la realiza el módulo responsable de las comunicaciones entre el interfaz de usuario y el modelo de resolución de problemas buscando en una tabla de relaciones pregunta-tarea.

En el mismo acto de formular la pregunta, bien por su expresión en lenguaje natural o bien porque son opciones de un menú, el usuario puede dar indicaciones sobre el tipo de respuesta que quiere recibir, por ejemplo, puede solicitar que ésta sea detallada y precisa, sin imponerle al sistema restricciones de tiempo, o puede pedir una solución aproximada pero obtenida en el menor tiempo posible. Estas indicaciones son tenidas en cuenta en la etapa de análisis



de la interacción en la que a la vista de: la pregunta actual y el tipo de usuario que la ha formulado, el diálogo mantenido recientemente entre el usuario y el sistema, y las preferencias del usuario sobre el modo de interactuar con el sistema, se estiman las expectativas del usuario respecto al diálogo. En función de estas expectativas se identifica un conjunto de criterios con los que explorar el ERP y que guiarán la selección de los métodos de resolución de problemas. Estos criterios estarán relacionados tanto con el contexto de información en el que debe obtenerse la respuesta como con requisitos de la interacción que debe llevarse a cabo con el usuario.

A partir de los requisitos de interacción y de la tarea de alto nivel asociada a la pregunta que hay que resolver, se diseña dinámicamente el modelo de tareas-métodos con el que se generará la respuesta apoyándose en conocimiento sobre las capacidades de resolución de problemas que proporciona la librería de métodos.

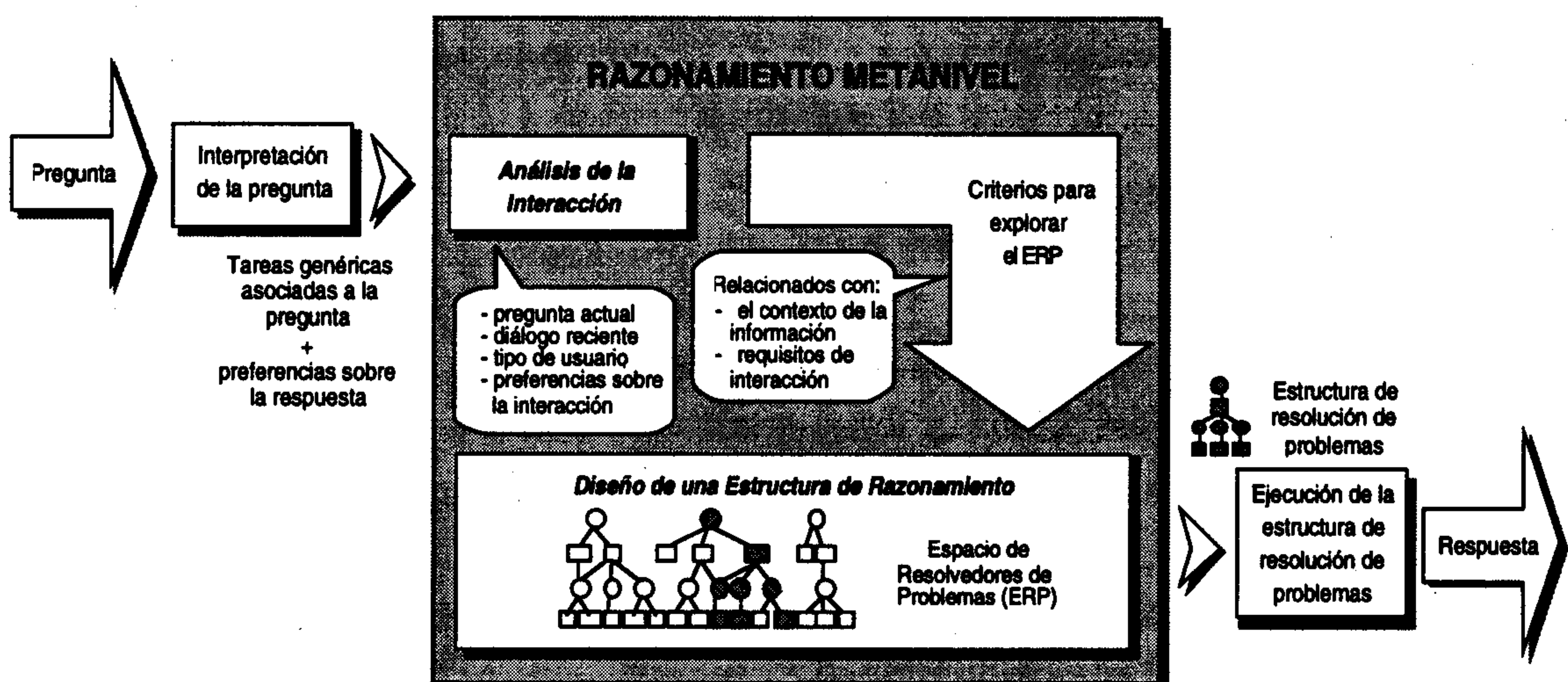


Figura 6.4: Esquema del razonamiento metanivel

Este modelo general de razonamiento está descrito en la figura 6.4 y sus aspectos más relevantes se describen en detalle en las secciones siguientes.

### 6.4.2 Modelo de escenarios de interacción

El objetivo de esta etapa es identificar el conjunto de requisitos de interacción que deben guiar la construcción del modelo de razonamiento. Estos requisitos se deducen de una caracterización del escenario de interacción en el que se



encuentran usuario y sistema, en el que un aspecto especialmente relevante es la caracterización del usuario y la intención que persigue con el diálogo, entendiendo como intención la actitud que adopta ante las posibilidades que le ofrece el sistema; por ejemplo, puede dialogar con el sistema para hacer consultas básicas sobre diversos aspectos del dominio o para controlar el estado de un área problema. La línea de razonamiento seguida consiste en analizar la clase de preguntas que ha hecho el usuario en las últimas etapas del diálogo y los atributos de interacción considerados en la generación de las respuestas para determinar qué tipo de usuario es, qué nivel de conocimientos tiene sobre el dominio problema y qué quiere del sistema. Una vez hecho esto, el modo en el que el sistema se comunicará con el usuario se define con un conjunto de requisitos de interacción que habrá que considerar para responder la pregunta actual y que se obtiene completando los requisitos expresados por el usuario al formular la pregunta con otros adecuados a sus características y al estado de la interacción.

#### **6.4.2.1 Atributos de interacción**

Los atributos de interacción definen características del diálogo entre un sistema y su usuario relativas a las prestaciones de los métodos de resolución de problemas. Toman valores sobre escalas cualitativas o dominios finitos de valores que definen la tendencia que debe guiar al sistema en la selección de los métodos. Como punto de partida y de forma general se han identificado los siguientes atributos de interacción, aunque ha de tenerse en cuenta que el conjunto de atributos de interacción empleados por un sistema es abierto, y por tanto admite la inclusión de otros, por ejemplo, dependientes del dominio:

- ♦ *Nivel de abstracción*: Relativo al grado de profundidad del método de razonamiento, esto es, al tipo de modelo de conocimiento del dominio sobre el que se apoya el método, por lo que cuanto más superficial sea el modelo más alto será el nivel de abstracción, y viceversa. La elección de un método con un nivel de abstracción alto implica que sus explicaciones serán superficiales, mientras que si el nivel de abstracción es bajo el método podrá ofrecer una justificación detallada de sus conclusiones. Normalmente, un mayor grado de abstracción va ligado a un menor tiempo de ejecución y viceversa.



- ◆ *Consumo limitado de recursos*: Relativo al uso que haga el método de recursos como el tiempo que tarda en ejecutarse, la memoria que emplea o por ejemplo, el número de conexiones remotas que utiliza. Por ejemplo, si el usuario busca una respuesta rápida entonces el valor de este atributo sería afirmativo.
- ◆ *Madurez de la información*. Relativo a la antigüedad/novedad de la información que se utiliza para generar las conclusiones. Si se elige la novedad el método preferido debe apoyarse en la información más reciente que se disponga; si por el contrario es la antigüedad entonces podría utilizarse un método que, por ejemplo, razone sobre información histórica.
- ◆ *Nivel de precisión*. Relativo al grado de detalle en el que se ofrecen las conclusiones del método. Por ejemplo, si un usuario le pide a un sistema que le diseñe un tratamiento médico a un nivel de precisión bajo, el sistema podría elegir un método que se limitase a sugerir los productos farmacéuticos genéricos que deben intervenir en el tratamiento, y el usuario, supuestamente un médico, se encargaría de detallar el tratamiento. Si por el contrario el usuario pide una respuesta con un nivel de precisión alto entonces el sistema elegiría un método alternativo que además de prescribir los productos anteriores detallase el número de tomas y las dosis necesarias de cada uno de ellos.
- ◆ *Consulta al usuario*. Puede haber métodos que para obtener resultados obtengan parte de la información directamente del usuario mediante la formulación de preguntas en tiempo de ejecución. La asignación de un valor negativo a esta característica haría que la selección se orientase hacia métodos que no realizan este tipo de consultas.
- ◆ *Nivel de asistencia*. Relativo a la cantidad de información que se proporciona en la respuesta. Puede haber métodos que ofrezcan más información de la estrictamente necesaria para constituir una respuesta aceptable; por ejemplo, un método podría hacer una predicción de la evolución del estado de un sistema aunque no se le hubiese solicitado explícitamente. Este es el caso de un método que analizando los datos sobre la situación actual de un área problema observa que ésta no es conflictiva, lo cual sería una respuesta suficiente para una pregunta sobre

el estado de dicha área, pero que complementa la respuesta con el pronóstico de problemas a corto plazo o con una llamada de atención hacia aspectos del entorno que pueden influir en la evolución de la situación.

- ◆ *Nivel de objetividad.* Relativo a la emisión de interpretaciones subjetivas o juicios de valor sobre las conclusiones del método. Por ejemplo, un método como el anterior que clasifica una situación a partir de un conjunto de datos de entrada podría limitarse a identificar la clase de situación en la que se encuentra el área problema o bien podría emitir un juicio sobre dicha situación basándose en el análisis de información histórica.
- ◆ *Valoración de resultados.* La confianza que tiene un método en la solución que ofrece puede estar graduada en función de la calidad o cantidad de los datos que recibe (como si están o no completos), o a que el método aplique técnicas de razonamiento con incertidumbre. En el caso en el que se utilicen métodos que emplean técnicas estadísticas, también podría ofrecerse una medida del error asociado a la respuesta.
- ◆ *Capacidad de explicación.* No todos los métodos tienen la capacidad de proporcionar explicaciones sobre la forma de obtener sus conclusiones, como es el caso de los métodos basados en procedimientos matemáticos. La asignación de un valor positivo a esta característica hará que la selección prime los métodos heurísticos sobre los matemáticos.
- ◆ *Múltiples soluciones.* Cuando se disponga de métodos capaces de generar más de una respuesta podría ofrecerse esta posibilidad al usuario a la vez que se le da la oportunidad de fijar el número máximo de las que querría recibir.
- ◆ *Flexibilidad en las respuestas.* Relativa a la actitud del método durante el proceso de razonamiento. Por ejemplo, un método cuya inferencia esté basada en un mecanismo de equiparación de patrones, puede realizar una equiparación bivaluada en la que la respuesta es simplemente positiva o negativa, o bien puede aplicar un modelo más flexible con el que se pueden ofrecer resultados parciales. Estos casos se corresponden con valores de la característica negativo y positivo, respectivamente.



- ♦ *Clase de métodos preferidos.* Partiendo de una clasificación sobre los métodos disponibles esta característica ofrece la posibilidad de fijar las preferencias sobre el tipo de métodos que deben utilizarse. Esta preferencia la puede manifestar el usuario directamente o bien la puede deducir el sistema en función de otras características del diálogo.

A la vista de los atributos de interacción anteriores puede observarse que, por lo general, no son independientes unos de otros. Por ejemplo, la imposición de restricciones de tiempo para la ejecución de un método irá ligada a la selección de métodos matemáticos o métodos heurísticos superficiales que son los más rápidos. Si además se solicitase la capacidad de generar explicaciones entonces la elección del método se decantaría hacia los métodos heurísticos. También la objetividad y el nivel de asistencia están relacionados, ya que si se buscan respuestas concisas (nivel de asistencia = bajo) entonces no parece coherente introducir juicios de valor sobre esas respuestas (nivel de objetividad = subjetivo).

Sin embargo, la definición de cada uno de estos atributos de forma independiente se justifica por la diversidad de los métodos de resolución de problemas. Puede haber métodos cuya caracterización tenga sentido para un subconjunto de estos modos, pero no para todos. Es más, en un escenario como el que se plantea en el ERP, en el que se dispone de múltiples métodos para llevar a cabo una misma tarea, las características reflejadas en los atributos no definen valores absolutos sino la posición que ocupa un método frente a otro respecto a dichas características. Por ejemplo, el nivel de abstracción de un método no es un concepto absoluto, es decir, no es alto o bajo *per se* sino que toma valor cuando se compara con otro método. Además, la lista de atributos anteriores puede reducirse o ampliarse en función de las necesidades de la librería de métodos: si la librería tiene un tamaño reducido podría bastar con un pequeño número de atributos, y a medida que se fuesen incorporando nuevos métodos este conjunto de atributos podría irse ampliando para caracterizar adecuadamente las prestaciones de unos métodos respecto a otros. La figura 6.6 muestra un ejemplo de asignación de dominios de valores a los atributos de interacción descritos anteriormente.

Por otra parte, la asignación de valores a los atributos de interacción no constituye una imposición sobre el tipo de diálogo que hay que entablar con el usuario sino la manifestación de una preferencia a la hora de elegir los métodos



que deben intervenir en la generación de las respuestas, que siempre estará supeditada a la disponibilidad de métodos con esas características en el ERP.

<b>NIVEL DE ABSTRACCION</b>	= bajo, medio, alto
<b>CONSUMO LIMITADO DE RECURSOS</b>	= sí, no
<b>MADUREZ DE LA INFORMACION</b>	= histórica, reciente
<b>NIVEL DE PRECISION</b>	= bajo, alto
<b>CONSULTA AL USUARIO</b>	= sí, no
<b>NIVEL DE ASISTENCIA</b>	= bajo, alto
<b>NIVEL DE OBJETIVIDAD</b>	= objetivo, subjetivo
<b>VALORACION DE RESULTADOS</b>	= sí, no
<b>CAPACIDAD DE EXPLICACION</b>	= sí, no
<b>MULTIPLES RESPUESTAS</b>	= sí, no (Nº Max: 1..10)
<b>FLEXIBILIDAD EN LAS RESPUESTAS</b>	= sí, no
<b>CLASE DE METODOS PREFERIDOS</b>	= matemáticos, heurísticos

Figura 6.5: Modos generales de interacción usuario-sistema

Además, si parte de estos atributos se hacen accesibles al usuario de manera que éste pueda elegir el valor que deben tomar para obtener la respuesta a su última pregunta, aumentan las posibilidades de adaptar adecuadamente la participación del sistema en el diálogo al ofrecer el usuario información concreta sobre sus preferencias, lo cual además contribuye a facilitar su caracterización. Es por ello que los atributos de interacción se han clasificado en dos grupos:

- aquellos a los que el usuario puede asignar un valor, que incluyen todos los que sean dependientes del dominio problema más los siguientes: nivel de abstracción, consumo limitado de recursos, madurez de la información, nivel de precisión, consulta al usuario y múltiples respuestas.
- aquellos cuyo valor sólo puede asignarlo el sistema: nivel de objetividad, valoración de resultados, capacidad de explicación, flexibilidad en las respuestas y clase de métodos preferidos.

En resumen, los atributos de interacción se utilizan con dos objetivos:



- (i) por una parte, caracterizar el tipo de diálogo usuario-sistema que se desea mantener con el usuario, y
- (ii) por otra, definir relaciones de orden parcial con respecto a una escala cualitativa de valores de características de un diálogo entre métodos capaces de resolver la misma clase de problemas.

### 6.4.2.2 Representación del conocimiento

La etapa de caracterización del escenario de interacción tiene como objetivo establecer el conjunto de requisitos de interacción que deben guiar el diseño del modelo de razonamiento con el que generar una respuesta. Este conjunto de requisitos ha de ser consistente con el tipo de usuario, su nivel de conocimientos sobre el dominio problema, y la intención con la que se comunica con el sistema. Una asignación de valores para estos tres últimos conceptos describe o etiqueta el tipo de interacción que se está manteniendo, y da una idea de los criterios que deberían tenerse en cuenta a la hora de ofrecer una respuesta.

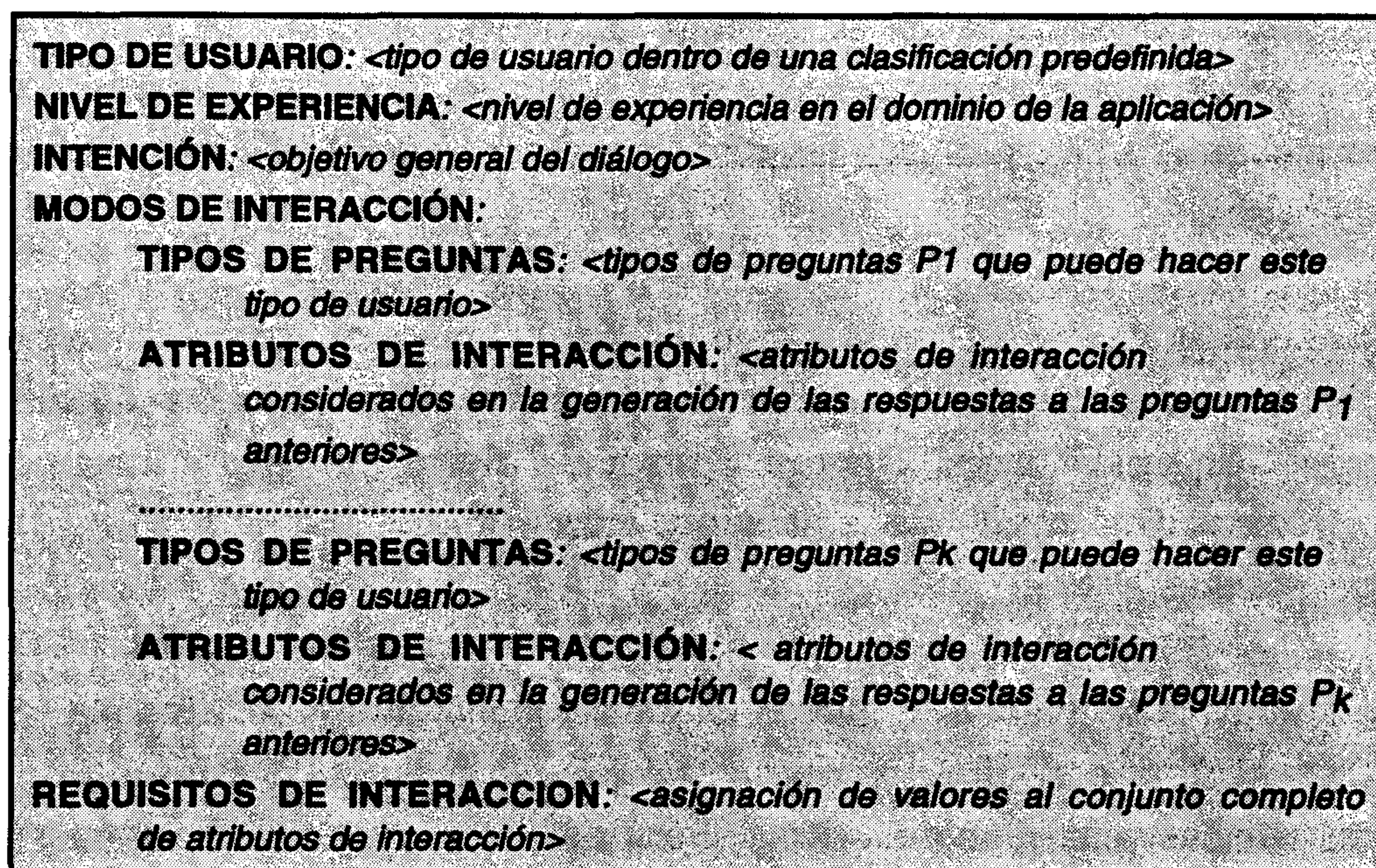


Figura 6.6: Caracterización general de un usuario

Por otra parte, la observación de las clases de preguntas formuladas por el usuario en momentos anteriores del diálogo junto con los atributos de



interacción que se emplearon para generar las respuestas puede sugerir características de la clase de usuario con quien se está dialogando y el objetivo de su consulta. Ahora bien, los atributos de interacción útiles para obtener estas características serán aquellos sobre los que el usuario tiene alguna capacidad de selección, por lo que son éstos los que se utilizarán para describir el diálogo reciente. Esta asociación entre características de usuario, clases de preguntas y atributos de interacción define un escenario de interacción, y se representa con marcos cuya estructura general es la que muestra la figura 6.6.

La descripción del escenario de la interacción del sistema con un interlocutor caracterizado por los conceptos *Tipo de Usuario*, *Nivel de Experiencia* e *Intención*, puede incluir varios grupos de asociaciones entre clases de preguntas y atributos de interacción ya que no necesariamente todas las preguntas se responden aplicando los mismos criterios de interacción. Por ejemplo, una sesión con un sistema de ayuda a la decisión en gestión de transporte público en la que el usuario es un operador experto que utiliza el sistema para mantener un nivel de servicio adecuado, estaría interesado en obtener propuestas de acciones de control detalladas que pueda evaluar e instalar sobre la red de transporte, pero le bastaría con una caracterización general de la situación ya que su experiencia le permite identificar rápidamente las peculiaridades de esa situación.

Por otra parte, si un usuario es clasificado como un operador experto con el objetivo de realizar la gestión completa de una de estas líneas de transporte público entonces sería razonable considerar que la siguiente respuesta que se le ofrezca se haya obtenido con la información más reciente de que se disponga y sea lo más precisa posible aunque incluyendo valoraciones sobre su calidad que el sistema sea capaz de justificar. Entonces, se entiende que para cada clase de usuario pueden definirse valores por defecto para el conjunto de los atributos de interacción que se especifican en la parte *Requisitos de Interacción* del marco (ver figura 6.6). Dicho conjunto de valores por defecto puede ser revisado antes de utilizarlo para hacer la selección de los métodos de resolución de problemas si el usuario, al hacer su consulta, ha asignado valores a todos o parte de los atributos de interacción que le son accesibles. En ese caso, las preferencias del usuario prevalecen sobre los valores del sistema. La figura 6.7 muestra el escenario de interacción con un usuario del tipo utilizado como ejemplo.



<b>TIPO DE USUARIO:</b> <i>operador</i>
<b>NIVEL DE EXPERIENCIA:</b> <i>experto</i>
<b>INTENCIÓN:</b> <i>gestión</i>
<b>MODOS DE INTERACCIÓN:</b>
<b>TIPOS DE PREGUNTAS:</b> <i>¿Qué está pasando en la línea &lt;L&gt;?</i> <i>¿Qué podría pasar en la línea &lt;L&gt; si se produjese un evento &lt;E&gt;?</i> <i>¿Qué debería hacerse en la línea &lt;L&gt; si se produjese un evento &lt;E&gt;?</i>
<b>ATRIBUTOS DE INTERACCIÓN:</b> <i>Nivel de abstracción = alto</i> <i>Consumo limitado de recursos = si</i> <i>Madurez de la información = reciente</i> <i>Nivel de precisión = bajo</i> <i>Consulta al usuario = no</i>
<b>TIPOS DE PREGUNTAS:</b> <i>¿Qué debería hacerse en la línea &lt;L&gt;?</i> <i>¿Por qué (qué debería hacerse en la línea &lt;L&gt;)?</i>
<b>ATRIBUTOS DE INTERACCIÓN:</b> <i>Nivel de abstracción = bajo</i> <i>Consumo limitado de recursos = no</i> <i>Madurez de la información = histórica</i> <i>Nivel de precisión = alto</i> <i>Consulta al usuario = si</i>
<b>REQUISITOS DE INTERACCIÓN</b> <i>Nivel de abstracción = bajo</i> <i>Consumo limitado de recursos = no</i> <i>Madurez de la información = reciente</i> <i>Nivel de precisión = alto</i> <i>Consulta al usuario = no</i> <i>Nivel de asistencia = bajo</i> <i>Nivel de objetividad = subjetivo</i> <i>Evaluación de resultados = si</i> <i>Capacidad de explicación = si</i> <i>Múltiples respuestas = si</i> <i>Flexibilidad en las respuestas = si</i> <i>Clase de métodos preferidos = heurísticos</i>

Figura 6.7: Ejemplo de escenario de interacción

Aunque aparentemente podría haber conflicto entre los atributos de interacción que desea un usuario y los que el sistema considera coherentes para un tipo de diálogo, este conflicto no llega a ser tal, ya que se considera que el rol de un usuario durante su diálogo con el sistema puede cambiar y la caracterización



de la interacción se hace teniendo en cuenta esta posibilidad. Por ejemplo, por el tipo de diálogo que ha mantenido el sistema con un usuario en las últimas tres preguntas, éste puede haber caracterizado el escenario de interacción como: el usuario es un operador con experiencia que quiere realizar la gestión del área problema; pero ante la forma en la que ha planteado la última pregunta el sistema puede cambiar de opinión y considerar que el usuario se está comportando como un operador sin experiencia y por tanto adecuar el modo en el que ofrecerá la respuesta a esta nueva actitud.

### **6.4.2.3 Caracterización del escenario de interacción**

La inferencia de los valores del conjunto de atributos de interacción que utilizará el razonador metanivel para configurar un modelo de razonamiento es el resultado de un proceso de equiparación sobre la base de conocimiento de marcos, cuyo resultado puede ser no determinista y tener asociado un grado de equiparación. Así, los escenarios de interacción pueden ordenarse de mayor a menor grado de equiparación, para aquellos que los tengan superior a cero, y pueden ser considerados por el sistema como opciones alternativas a considerar en la etapa de diseño del modelo de razonamiento. La disponibilidad de distintas soluciones u opciones haría posible que el sistema emplease algún tipo de método de evaluación de la calidad de la solución que ante un resultado negativo diese la posibilidad de reconstruir el modelo de razonamiento considerando un escenario de interacción alternativo, es decir, un conjunto de atributos de interacción diferente.

En líneas generales, el proceso de equiparación tendría las siguientes características:

- Se observan las últimas  $n$  preguntas (el valor de  $n$  es modificable) formuladas por el usuario y los valores de los atributos de interacción accesibles por el usuario que fueron utilizados para generar las correspondientes respuestas así como los que definen las preferencias del usuario con respecto a la forma de proporcionarle la respuesta a su pregunta actual. Las preguntas consideradas pueden ser tanto clases de preguntas, como se muestra en el ejemplo de la figura 6.7, como instancias de alguna clase.



- Se analiza el grado de equiparación de la sección *modos de interacción* de los marcos propios del tipo de usuario con el que se está interactuando, y éste determina el grado de equiparación del marco. El tipo de usuario se conoce desde el inicio de la sesión puesto que se entiende que antes de comenzar, el usuario ha de identificarse y el sistema dispone de la información sobre la clase a la que pertenece cada uno de los usuarios con los que puede comunicarse.

El nivel de equiparación de esta sección se estimará en dos pasos:

- (i) primero se evalúa el grado de equiparación de los distintos grupos de preguntas y su conjunto de valores para atributos de interacción asociados, y
  - (ii) después se combinan los niveles de equiparación obtenidos en el paso anterior para obtener el grado de equiparación total.
- Finalmente, se ordenan los marcos en función de su grado de equiparación y se toman los valores de los atributos de interacción incluidos en la sección de *requisitos de interacción* del marco que haya alcanzado mayor grado de equiparación.

En resumen, el método de caracterización del escenario de interacción descrito plantea un tipo de enfoque, no un procedimiento preciso, y por tanto está abierto a la incorporación de diferentes soluciones con las que especificar las distintas etapas.

### 6.4.3 Modelo reflexivo de resolución de problemas

Una vez establecidos los criterios de interacción que deben guiar la comunicación del sistema con su usuario, queda por ver cómo van a utilizarse éstos para explorar la librería de métodos seleccionando aquellos cuyas prestaciones se acercan más a las que se solicitan. Además de los criterios de interacción, la valoración de un método como más adecuado que otros para resolver un mismo problema depende de otros factores relativos al contexto en el que se plantea la utilización de estos métodos. Estos factores pueden referirse a características de la información de entrada a los métodos, o a la disponibilidad de determinado conocimiento en la memoria de la interacción.

Esta memoria se construye durante el desarrollo del diálogo usuario-sistema con la información que se va generando como resultado de los distintos procesos de razonamiento aplicados para responder preguntas. Entonces, la decisión en tiempo de ejecución sobre el método que debe aplicarse para resolver una tarea estará condicionada por:

- (i) la evolución del diálogo y su estado actual, representado por los requisitos de interacción, y
- (ii) el estado de conocimiento en el que se encuentre el sistema.

Las consideraciones anteriores imponen requisitos sobre el modelo de razonamiento que finalmente se emplee, por lo que parece adecuado abordar la tarea de construir dicho modelo como la resolución de un problema de diseño. Concretamente, este clase de problema de diseño es el que Brown y Chandrasekaran, [Brown, Chandrasekaran, 89], [Chandrasekaran, 90], denominaron *diseño rutinario*, y la solución presentada en esta tesis emplea una versión adaptada, de su modelo de resolución de problemas, al problema de razonar sobre conocimiento metanivel bajo restricciones de interacción usuario-sistema. En las secciones siguientes se muestra cómo se expresa ese conocimiento de diseño y cómo se utiliza para construir un modelo de razonamiento con el que generar la respuesta que espera el usuario.

#### **6.4.3.1 Representación del conocimiento**

El tipo de problemas clasificados como de diseño rutinario son aquellos dirigidos a construir un artefacto con ciertas funcionalidades sujetas a un conjunto de restricciones. El resolvedor de problemas de diseño de esta clase tiene como objetivo alcanzar el diseño final mediante la descomposición del problema de diseño en subproblemas más sencillos. Para ello se vale de una colección jerárquica de especialistas de diseño, en cuyos niveles superiores residen los especialistas en los aspectos más generales del objeto a diseñar, mientras que los de los niveles inferiores trabajan sobre subproblemas o componentes más específicos. Los especialistas elegidos para un sistema de diseño, sus responsabilidades y su organización jerárquica reflejan la estructura conceptual del dominio problema que usan los diseñadores.



El conocimiento del que disponen estos especialistas son planes de diseño, que representan una solución parcial precompilada de un objetivo de diseño, y que se definen como una secuencia de acciones de diseño con la que se especifica cómo descomponer el problema inicial en subproblemas más sencillos. Cada una de estas acciones representa la aplicación de conocimiento de diseño local al especialista o una llamada a otro especialista de un nivel inferior en la jerarquía. Como un especialista puede disponer de varios planes de diseño se hace necesario un proceso de selección cuyas decisiones dependen de tres clases de información:

- (i) la calidad del plan en sí mismo,
- (ii) las demandas o requerimientos iniciales del usuario, y
- (iii) la situación en la que la selección tiene lugar, es decir, el estado del diseño y su historia.

Entonces, puede verse el proceso de diseño rutinario como un proceso de refinamientos sucesivos de un plan general de diseño llevados a cabo por especialistas que insertan gradualmente planes más específicos dentro del plan general hasta conseguir construir el plan con el nivel de detalle necesario para producir el diseño deseado.

MODELO DE DISEÑO RUTINARIO	MODELO REFLEXIVO METANIVEL
<i>Objetivo.</i> Diseñar un artefacto que proporcione ciertas funcionalidades satisfaciendo un conjunto de restricciones de diseño	<i>Objetivo.</i> Diseñar un modelo de razonamiento capaz de generar una respuesta adaptada a los requisitos de la interacción usuario-sistema
Especialistas	Tareas
Planes de diseño	Métodos de resolución de problemas
Descomposición de planes en subobjetivos	Descomposición de métodos en subtareas
Selección de planes en función de: <ul style="list-style-type: none"><li>• calidad de los planes</li><li>• requerimientos del usuario</li><li>• estado del diseño</li></ul>	Selección de métodos en función de: <ul style="list-style-type: none"><li>• características de los métodos</li><li>• requisitos de interacción</li><li>• estado de conocimiento del sistema</li></ul>

Figura 6.8: Análisis comparativo del modelo de diseño rutinario y el modelo de razonamiento reflexivo metanivel



Al aplicar este modelo sobre el problema de diseñar dinámicamente una estructura de conocimiento que genere la información solicitada por un usuario, sus principales elementos quedan caracterizados como sigue (ver figuras 6.8, 6.9):

- Hay un único especialista por cada una de las tareas que pueden realizarse en el ERP. La responsabilidad de estos especialistas es identificar cuál de los métodos disponibles para realizar la tarea es el más adecuado.
- Los planes de diseño de un especialista son los métodos de resolución de problemas de que dispone para llevar a cabo la tarea, y la descomposición de los mismos se corresponde con las subtareas de estos métodos.

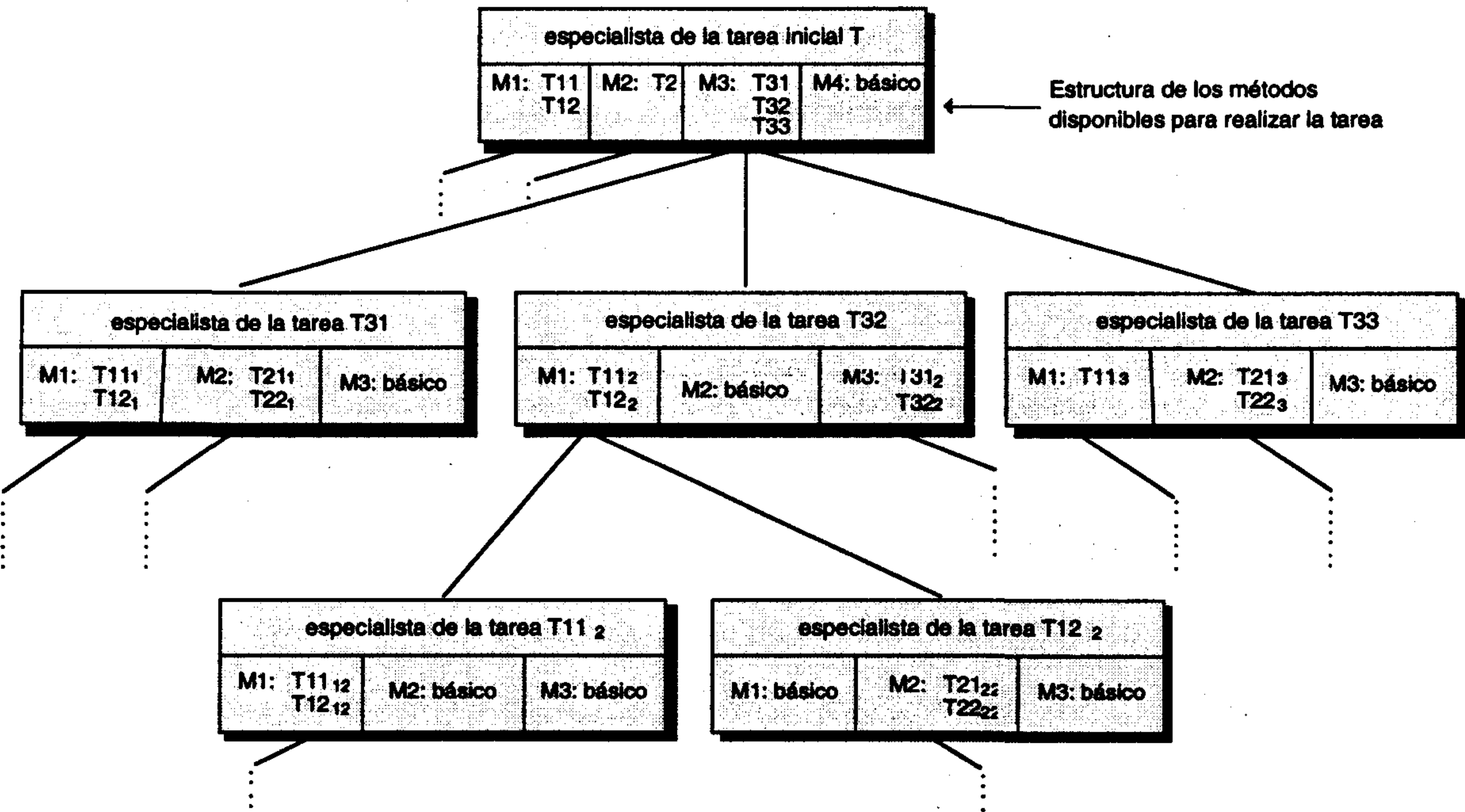


Figura 6.9: Modelo de diseño rutinario aplicado a la construcción de un modelo de razonamiento

Por analogía con el tipo de información considerada por el selector de planes del modelo general, el conocimiento de selección de métodos se apoya en:

- las características de los propios métodos,



- los requisitos de interacción expresados por el usuario y los deducidos por el sistema y,
- el estado de conocimiento del sistema en el momento de hacer la selección.

La representación del conocimiento de selección de métodos se realiza con reglas cuya parte izquierda incluye una expresión formal de hipótesis relativas a:

- el valor de un atributo de interacción,
- una propiedad de la información de entrada a un método, o
- una creencia sobre información de contexto; y cuya parte derecha incluye el nombre del método.

Naturalmente, todos los especialistas disponen al menos de un método, y todos ellos tienen un método por defecto que es el que se elige si las condiciones para aplicar los demás métodos no se satisfacen.

Este conjunto de reglas de selección de métodos se considera ordenado, buscando con ello simular el proceso de selección que llevaría a cabo una persona en circunstancias similares entendiendo que en esos casos la persona primaría el análisis de las condiciones de aplicabilidad de unos métodos antes que de otros, ya sea porque estos métodos tienen unas características especiales o porque la evaluación de sus condiciones es más rápida.

Los atributos de interacción que se incluyen en la especificación de estas condiciones son solamente aquellos relevantes para el método en cuestión. Por ejemplo, supóngase que se dispone de tres métodos para realizar la tarea de clasificar una situación a partir de un conjunto de datos de entrada:

- (i) un método heurístico que equipara los datos recibidos con patrones descriptivos de situaciones prototípicas,



- (ii) otro método heurístico similar al anterior que además, durante el análisis de los datos, extrae aspectos de la situación actual que podrían indicar su posible empeoramiento en un futuro próximo, y
- (iii) un método matemático que emplea un modelo estadístico para identificar la situación.

Al plantearse la selección de uno de estos métodos, la elección entre el primero y el segundo podría girar en torno al valor del atributo de interacción nivel de asistencia, si éste es bajo se optaría por el primero y si no por el segundo. Entre el segundo y el tercero, o el primero y el tercero, el nivel de asistencia ya no es el factor discriminante entre los métodos, sino que habría que considerar otros atributos como la capacidad de dar explicaciones, el consumo limitado del recurso tiempo (que suele ser menor en el caso de métodos matemáticos) o la precisión de la respuesta.

La figura 6.10 muestra el esquema general de representación del conocimiento de estos especialistas.

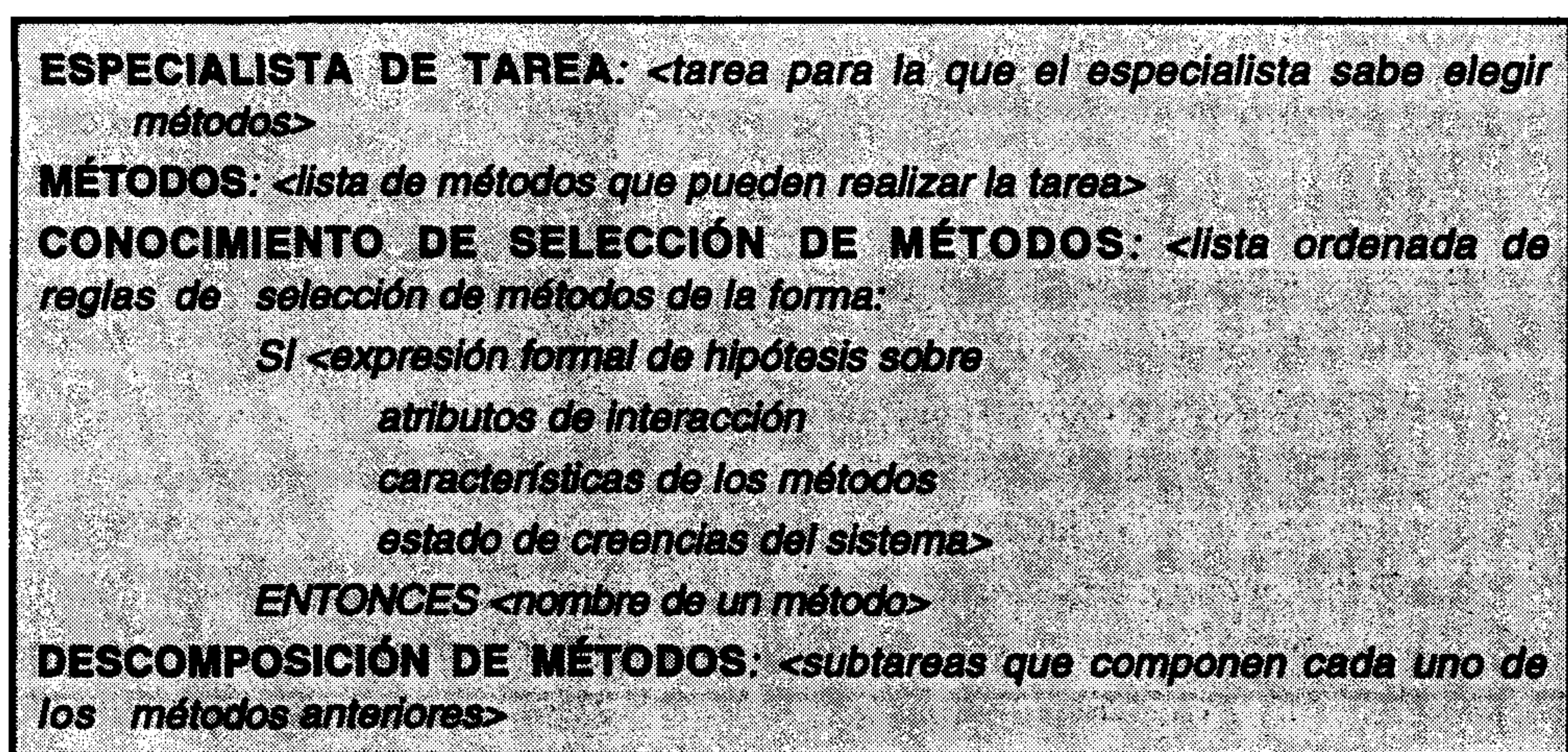


Figura 6.10: Especialista en selección de métodos para una tarea

La figura 6.11 muestra un ejemplo de especificación de un especialista en la tarea de planificar acciones de control sobre un área problema. Para ello se dispone de tres métodos, los cuales reciben como entrada la situación observada en el área:



- un método de *planificación directa* que a la vista de la situación actual deduce qué planes de control podrían llevarse a cabo,
- un método similar al anterior, llamado *planificación con alerta*, que además de sugerir planes de acciones a tomar hiciese una llamada de atención sobre la efectividad de los mismos basada en la presencia prevista de determinados eventos en el área problema, y
- un método más complejo, denominado *predicción & planificación*, que primero predice la evolución a corto plazo de la situación actual para luego diseñar un plan de actuación detallado adecuado a esa evolución.

**ESPECIALISTA DE TAREA:** *planificar acciones de control*  
**MÉTODOS:** *planificación directa, planificación con alerta, predicción & planificación*  
**CONOCIMIENTO DE SELECCIÓN DE MÉTODOS:**  
*Si nivel de abstracción = alto  $\wedge$  nivel de asistencia = bajo*  
**ENTONCES** *planificación directa*  
  
*Si nivel de abstracción = alto  $\wedge$  nivel de asistencia = alto  $\wedge$  pertenece(situación observada, problemas graves)*  
**ENTONCES** *planificación con alerta*  
  
*Si pertenece(situación observada, problemas leves)*  
**ENTONCES** *planificación directa*  
  
**EN OTRO CASO** *predicción & planificación*  
**DESCOMPOSICIÓN DE MÉTODOS:**  
*planificación directa: decidir acciones de control*  
*planificación con alerta: decidir acciones de control, identificar alertas*  
*predicción & planificación: predecir situación, decidir acciones de control*

Figura 6.11: Ejemplo de especialista en una tarea de planificación

Los aspectos que primero se analizan son los niveles de abstracción y asistencia que requiere la interacción con el usuario. Los métodos de *planificación directa* y *planificación con alerta* pueden ofrecer ambos una respuesta a un nivel de abstracción alto, pero el segundo proporciona más información que el primero, aunque su uso es adecuado cuando la situación observada es un problema grave. Por otra parte, la gestión de los problemas calificados como leves no



requiere estimar cómo van a evolucionar éstos por lo que, en estos casos, el método de planificación directa se considera suficiente. Finalmente, si no se dan las condiciones anteriores, el método que se seleccionaría sería el de *predicción & planificación*.

### **6.4.3.2 Diseño de un modelo de razonamiento**

Al aplicar el conocimiento anterior para diseñar un modelo de razonamiento se busca construir una estructura de inferencia en la que queden determinados cada uno de los pasos a seguir para llegar a generar la respuesta que ha de ofrecerse al usuario. La línea general del razonamiento con el que alcanzar este objetivo consiste en:

- (1) Como punto de partida, se identifica al especialista en la tarea de alto nivel obtenida de la traducción de la pregunta del usuario al lenguaje de resolución de problemas.
- (2) El especialista analiza la satisfacibilidad de las condiciones expresadas en las reglas de selección de métodos de resolución de problemas en el orden en el que han sido definidas. El método seleccionado será aquel asociado a la primera regla para la que se satisfagan las condiciones de su parte izquierda salvo que el especialista sólo disponga de un método con el que llevar a cabo la tarea, en cuyo caso éste es el método elegido. El antecedente de una regla, expresado como una conjunción de condiciones, se satisface sólo si lo hacen todas las condiciones que lo definen lo cual puede suponer:
  - comparar los valores de los atributos de interacción incluidos en ese antecedente con los que especifican los requisitos de interacción determinados en la etapa de caracterización de la interacción,
  - analizar algunas características o requisitos de las entradas de los métodos
  - acceder a la memoria de la interacción para consultar sobre aspectos del conocimiento generado en etapas anteriores del diálogo.



- (3) Si el método seleccionado es un método básico entonces el diseño de la parte del modelo de razonamiento en la que intervenía este especialista se da por concluida.
- (4) Si el método seleccionado es compuesto se repetirá el proceso, volviendo al punto (2), con cada uno de los especialistas de las subtarear del método hasta conseguir la estructura completa.

Durante el desarrollo de este proceso se irá construyendo una jerarquía de tareas y métodos, producto de las decisiones de diseño anteriores, que tras la intervención de cada especialista se irá refinando y especificando hasta conseguir un modelo al mayor nivel de detalle posible, cuyos niveles inferiores son ocupados por métodos básicos. El grado de detalle que tenga el modelo de razonamiento construido será el máximo que permite el conocimiento incluido en el ERP.

## 6.5 El Contexto de la Interacción

A la vista del tipo de diálogos que desean mantenerse entre usuarios y sistemas, surge la necesidad de incorporar una memoria del estado de conocimiento de la aplicación para poder gestionar de forma eficiente la generación de las diferentes respuestas. La disponibilidad de una memoria de la evolución del diálogo usuario-sistema en la que se incluyan tanto los resultados de los distintos procesos de resolución de problemas que dieron lugar a las respuestas como aspectos relevantes asociados a la generación de esas respuestas, si es gestionada adecuadamente, presenta una serie de claras ventajas de cara a mejorar el nivel de comunicación entre usuarios y sistemas:

- ⇒ Los procesos de inferencia aplicados para responder preguntas que se plantean como alternativas a preguntas formuladas con anterioridad, en las que se introducen hipótesis que modifican aspectos del comportamiento o estado del sistema, se apoyan en gran parte de la información generada para responder a dichas preguntas. El acceso directo a esta información, contenida en la memoria, proporciona *agilidad* al diálogo al disminuir el tiempo de respuesta del sistema.

⇒ Entendiendo esta memoria como una representación del estado de creencias del sistema, si se dispone de un mecanismo gestor de la memoria que pueda mantener consistente dicho conjunto de creencias, incorporando nuevas creencias y revisando aquellas que han pasado a ser falsas, se consigue *flexibilidad* en el desarrollo de los diálogos al no hacer que estos evolucionen según un guión rígido, más o menos preestablecido, sino que de forma similar a como sucede en las conversaciones entre personas, se da la posibilidad de formular preguntas que van añadiendo hipótesis de forma sucesiva, reconsiderando hipótesis planteadas previamente, o declarar incompatibilidades entre hipótesis que hacen falsas las conclusiones extraídas a partir de ellas y que por tanto requieren una revisión.

Entonces, por ejemplo, en general podría darse soporte a preguntas como *¿qué pasaría si H1?* o *¿qué habría que hacer si H2?*, pero también puede enriquecerse el tipo de diálogo mediante el uso de etiquetas lingüísticas en la formulación de las preguntas. La presencia de estas etiquetas influye en la interpretación de las preguntas, haciendo que a la hora de razonar para generar la correspondiente respuesta dichas hipótesis sean consideradas de forma conjuntiva o disyuntiva en función de la estructura de la pregunta.

Además, a la hora de utilizar las etiquetas, podría incrementarse la naturalidad de los diálogos usuario-sistema ofreciendo la posibilidad de que todas las hipótesis a considerar en el razonamiento no estén declaradas de una vez en la pregunta sino que puedan declararse en diferentes consultas, haciendo que la conclusión sobre su carácter conjuntivo o disyuntivo se decida en función de la estructura de la pregunta. Por ejemplo, si a la pregunta *¿qué pasaría si H1?* le sigue *¿qué pasaría si H2?* el efecto sobre el estado de creencias del sistema debería ser el mismo que produce la pregunta *¿qué pasaría si H1 o H2?*, ya que en definitiva en ambos casos hay que resolver la misma clase de problemas, que es obtener dos descripciones del estado de un sistema: el que resulta de asumir la hipótesis H1 y por otra parte, el que se obtiene suponiendo H2; aunque en distintos plazos de tiempo. Así mismo, si a *¿qué pasaría si H1?* le sigue la pregunta *¿y si también sucediese H2?* se entendería que el estado del que se busca una descripción debería ser aquel obtenido suponiendo ambas hipótesis simultáneamente, lo que supondría volver a resolver la tarea asociada al tipo de pregunta *qué pasaría si* pero esta vez considerando las dos hipótesis, o



lo que es lo mismo, significaría reformular la segunda pregunta como *¿qué pasaría si H1 y H2?*.

Por otra parte, si se plantea la flexibilización del diálogo dando la posibilidad de reconsiderar hipótesis planteadas con anterioridad, y por tanto las conclusiones derivadas de ellas, han de habilitarse los mecanismos para poder revisar el conjunto de creencias del sistema de manera que se invaliden aquellas conclusiones que ya no están soportadas por hipótesis consistentes. Por ejemplo, tras una pregunta del tipo *¿qué habría que hacer si H1, H2 y H3?*, el usuario podría darse cuenta de que H1 y H3 no pueden coexistir por lo que podría hacerle al sistema la declaración *H1 y H3 son inconsistentes*. Esto debería provocar, no sólo que se invalidasen las conclusiones alcanzadas en la pregunta anterior, sino también todos aquellos resultados apoyados por las hipótesis H1 y H3 que pudiesen haberse obtenido previamente.

Las consideraciones anteriores hacen adecuado el uso de un modelo de gestión de las creencias del sistema definido como un Sistema de Mantenimiento de la Verdad (*Truth Maintenance System, TMS*). Un TMS tiene por objeto, cuando opera asociado a un resolutor de problemas, facilitar los trabajos de comprobación de la veracidad de las condiciones necesarias para llevar a cabo los procesos de inferencia del resolutor. Esta actividad presenta dos características a tener en cuenta:

- (i) por una parte, si se trabaja con procesos complejos esta operación de verificación resulta muy costosa ya que el número de declaraciones a revisar es muy numeroso y,
- (ii) por otra parte, cuando el mundo descriptivo de un estado es complejo el alcance de las modificaciones introducidas por la aplicación de un proceso de inferencia suele estar restringido, manteniendo inalterado gran parte del conjunto de creencias.

Ante estos hechos y para facilitar dicha tarea, el TMS se encarga de mantener consistente el conjunto de creencias del sistema, introduciendo nuevas creencias y eliminando las que dejan de ser ciertas como resultado de la aplicación de algún proceso de inferencia. De esta manera se consigue un comportamiento no monótono ya que el TMS se ocupa de mantener eficientemente el conjunto de creencias relevantes para el resolutor de

problemas en forma consistente con los cambios que el proceso de inferencia ha introducido en el dominio de valores de las variables de situación. Por ello, el mundo correspondiente a un estado es un ejemplo de mundo provisional en el que se asumen primero unas creencias que luego se retraen como consecuencia del efecto de los operadores aplicados a ese mundo.

Entonces, de forma general, un TMS puede definirse como un sistema que mantiene un conjunto de hipótesis y justificaciones de forma que es capaz de responder sobre la validez de una o varias afirmaciones en el contexto del que entiende y es capaz de actualizar sus creencias en función de las modificaciones aportadas por el usuario del TMS, el cual puede ser tanto un usuario humano o un proceso de resolución de problemas.

### 6.5.1 Representación del conocimiento

El tipo de TMS empleado para gestionar la memoria de un diálogo usuario-sistema es el ATMS (*Assumption-Based Truth Maintenance System*) [De Kleer, 86]. El ATMS se caracteriza por manejar de forma explícita un retículo de combinaciones de hipótesis, llamadas entornos, y de nodos deducibles de esas combinaciones de manera que es posible establecer rápidamente la consistencia de los entornos y a partir de ellos la validez de los distintos nodos. En concreto, los elementos más significativos del modelo ATMS son los siguientes:

- un *nodo* del retículo describe una creencia del sistema y constituye un dato utilizado o generado por el resolutor de problemas. Cabe distinguir tres tipos de nodos: (i) premisas, que representan hechos o datos reales considerados siempre válidos, (ii) hipótesis, que representan datos hipotéticos cuya validez es revisable y, (iii) nodos deducidos, que representan datos obtenidos por el resolutor de problemas a partir de un conjunto de premisas y/o hipótesis.
- una *justificación* describe la forma de derivar un nodo a partir de otros.
- un *entorno* es un conjunto de hipótesis, asumidas como válidas, que se entiende como la conjunción de las mismas.
- un *contexto* es el conjunto formado por un conjunto de nodos hipótesis más los nodos derivables de éstos y las justificaciones aplicables.



Un ATMS presenta fundamentalmente dos funcionalidades:

- (i) Determina a partir de un conjunto de hipótesis el contexto asociado a dicho conjunto de forma eficiente, teniendo en cuenta que al recibir las justificaciones por parte del resolutor de problemas de forma incremental los nuevos contextos pueden obtenerse de la actualización de los ya disponibles, sin necesidad de generarlos nuevamente.
- (ii) Da soporte al resolutor de problemas para saber si ante nuevas hipótesis alguno de los contextos se hace inconsistente o si determinada información necesaria para llevar a cabo una inferencia puede deducirse de algún contexto consistente. Se dice que un contexto es inconsistente si contiene una contradicción, es decir, si hace que coexistan dos hipótesis que son incompatibles. Una contradicción puede obtenerse o bien porque ha sido definida por el usuario como una restricción de integridad o bien puede resultar de la inferencia de dominio nulo de valores para algún nodo derivable.

Con estos objetivos, el ATMS maneja una estructura de datos en la que se relacionan directamente nodos con entornos de hipótesis mediante el concepto de *etiqueta*. La etiqueta de un nodo es el conjunto de entornos mínimos a partir de los cuales puede deducirse el nodo aplicando el conjunto de justificaciones. Por ejemplo, en el caso de las premisas, como éstas reflejan hechos o datos no obtenidos a partir de ningún otro su etiqueta está vacía.

Entonces, a la vista de las definiciones anteriores, la caracterización completa de un nodo incluirá, además de la descripción o expresión del nodo, su etiqueta y sus justificaciones. Teniendo esto en cuenta, los tres tipos de nodos descritos anteriormente se caracterizarían porque:

- los nodos premisa tienen etiqueta y conjunto de justificaciones vacíos ya que reflejan hechos no obtenidos a partir de ningún otro,
- los nodos hipótesis incluyen en su etiqueta un entorno que contiene a la propia hipótesis, y
- los nodos deducidos representan el caso general en que pueden darse todos los casos de etiqueta y conjunto de justificaciones no vacíos.

En el contexto de esta tesis, en el que se plantea el uso de un modelo ATMS para gestionar la memoria de un diálogo entre un sistema y su usuario, los elementos descritos anteriormente quedarían caracterizados como sigue:

- Un nodo premisa será cualquier dato proporcionado por un sistema de información o declarado como un hecho por el usuario.

Un nodo hipótesis será cualquier dato hipotético declarado por el usuario al formular una pregunta. Al considerar estas hipótesis dentro del contexto de un sistema de ayuda a la decisión, dichas hipótesis se referirán en general a condiciones del entorno, estado o acciones de control relativas al sistema que se está monitorizando.

Un nodo deducido será cualquier información generada por el sistema tras aplicar un modelo de razonamiento basado tanto en nodos premisa como en nodos hipótesis o deducidos.

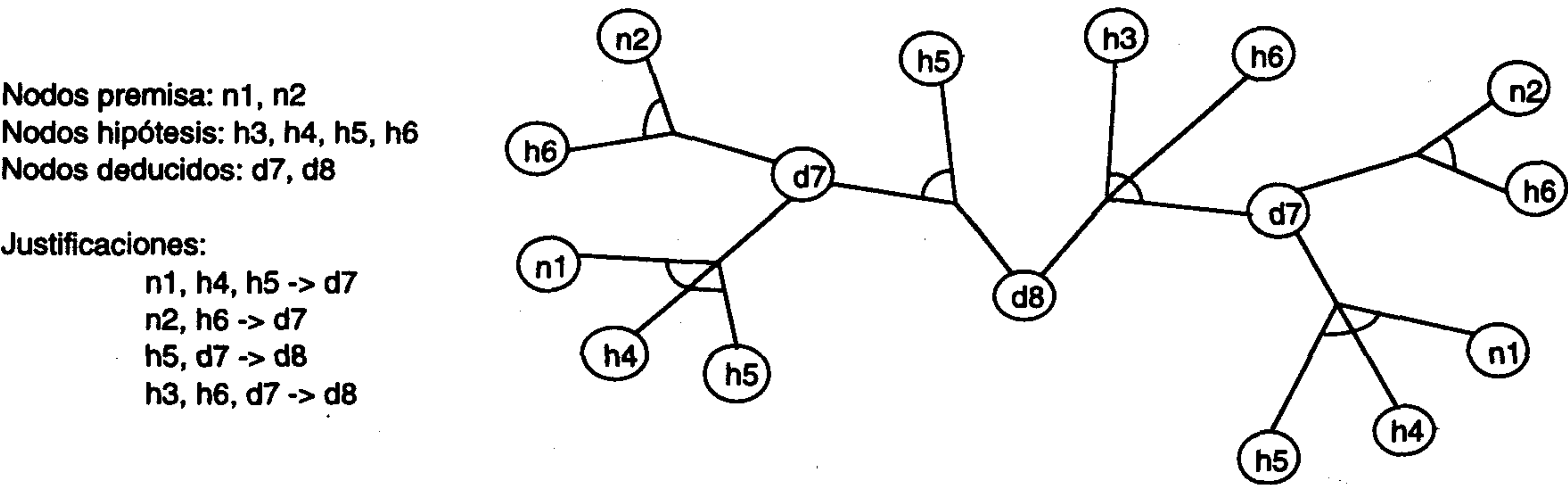
- Una justificación es una pieza de la traza del razonamiento con el que el resolutor de problemas ha obtenido una serie de conclusiones o respuestas. Cada vez que se aplica un proceso de inferencia, el grado de detalle con el que está definida esta traza, es decir, el número de nodos utilizados para describir cada resultado y el número de justificaciones aportadas para describir la línea de razonamiento seguida, depende de los términos en los que está especificado el tipo de diálogo que desea mantenerse con el usuario. Cuantas más posibilidades se le dé al usuario para formular preguntas basadas en datos hipotéticos, mayor número de nodos habrá que incluir en el retículo y por tanto mayor número justificaciones que apoyen esos nodos.

Por otra parte, la aplicación de un proceso de inferencia genera además de las conclusiones finales, una serie de resultados intermedios entre los que cabe distinguir aquellos que podrían ser utilizados para resolver problemas posteriores y los que no. La identificación de los primeros determina el conjunto de justificaciones que ha de aportar el resolutor de problemas al conjunto de creencias cada vez que aplique dicho proceso de inferencia. Esta distinción es necesaria para mejorar la gestión de la memoria, evitando sobrecargarla de información que no va a ser



utilizada en la gestión de la interacción y que puede ser recogida en otro tipo de soporte, como una base de datos o un fichero.

- Los conceptos de entorno y contexto se mantienen inalterados con respecto al modelo general.



Nodo	Descripción	Etiqueta	Justificaciones
n1	<obj1, atr1, v1>	{ }	(n1)
n2	<obj2, atr2, v2>	{ }	(n2)
h3	<obj3, atr3, v3>	{h3}	(h3)
h4	<obj4, atr4, v4>	{h4}	(h4)
h5	<obj5, atr5, v5>	{h5}	(h5)
h6	<obj6, atr6, v6>	{h6}	(h6)
d7	<obj7, atr7, v7>	{h4, h5} {h6}	(n1, h4, h5) (n2, h6)
d8	<obj8, atr8, v8>	{h4, h5} {h5, h6} {h3, h6}	(h5, d7) (h3, h6, d7)

Figura 6.12: Representación del estado de creencias de un sistema  
(adaptada de [Cuenca, 97])

Todo nodo del retículo - premisa, hipótesis, o deducido, es una proposición que se expresa formalmente como una terna <objeto, atributo, valor>. Dado que el usuario al hacer una pregunta puede formular hipótesis que plantean valores alternativos de un mismo concepto, y que por otra parte el sistema puede generar distintos valores para un mismo concepto al razonar bajo diferentes hipótesis, el retículo puede contener varias ternas diferentes para el



mismo par *<objeto, atributo>*. Esto hace que puedan identificarse tipos de nodos de los que, en principio, podría llegar a haber tantos ejemplares en el retículo como determine el dominio de valores del par *<objeto, atributo>* que define el tipo.

Los diferentes tipos de nodos que pueden llegar a formar parte del estado de creencias de una sistema dependen del conjunto de métodos que haya disponible en el Espacio de Resolvedores de Problemas. Para cada método es necesario destacar cuáles son los conceptos que puede aportar al retículo, lo que significa que ha de identificarse qué clase de información, datos o hipótesis, puede recibir el método como entrada y qué clase de información, deducciones, puede ofrecer como salida, todo ello expresado con pares *<objeto, atributo>*. Los conceptos seleccionados para formar parte del retículo serán solamente aquellos que pueden tener relevancia en el desarrollo de un diálogo usuario-sistema, ya sea porque son conceptos que en un momento dado el usuario pueda emplear para comunicarse con el sistema, que el sistema use para expresar información o de los que se tenga información a priori sobre sus relaciones de inconsistencia con otros conceptos.

Por otra parte, las justificaciones que proporcionen estos métodos de resolución de problemas, aunque pueden definirse a priori para clases de nodos, serán especificadas tras la ejecución de los métodos y se formarán sólo con los extractos de la traza del proceso de inferencia del método en la que aparecen conceptos asociados a nodos.

El estado del retículo de creencias del sistema puede representarse en una tabla en la que cada nodo tiene asociadas su etiqueta y sus justificaciones, como la que se muestra en el ejemplo de la figura 6.12. A continuación se describe cómo se actualiza el conjunto de creencias con el transcurso de un diálogo usuario-sistema.

### 6.5.2 Gestión de la memoria

Al entender que el conjunto de creencias de un sistema constituye una memoria del diálogo que ha mantenido con un usuario, se plantean problemas propios de la gestión de memoria: cómo añadirle nueva información, cómo acceder a la información que contiene, cómo revisar su consistencia y cómo



mantener su tamaño dentro de unos límites razonables permitiéndole que olvide información.

### 6.5.2.1 Adición de información

La obtención de la respuesta a una pregunta de un usuario implica la ejecución de un proceso de resolución de problemas que genera nueva información - nuevos nodos con sus justificaciones o nuevas justificaciones para un mismo nodo- que ha de añadirse al conjunto de creencias del sistema de manera que el estado final de conocimiento del sistema se mantenga consistente. El procedimiento mediante el cual se lleva a cabo la actualización de la estructura de nodos al añadir una justificación tiene tres pasos fundamentales:

- (1) Se computa una nueva etiqueta para el nodo justificado uniendo de todas las formas posibles los entornos de las etiquetas de los nodos justificativos y los que ya pudiese tener el nodo. Por ejemplo, si el estado de creencias sobre un nodo N es:

$$\text{Etiqueta} = \{Ng, Nr, Nh\} \quad \text{Justificaciones} = (Nf, Nh)$$

y se introduce una nueva justificación para N:

$Na, Nb \rightarrow N$  donde

Etiqueta de Na:  $\{Nr, Nh\} \{Np, Nq\}$

Etiqueta de Nb:  $\{Ne, Nf\} \{Nk, Nh\}$

la nueva etiqueta de N saldría de la combinación de los entornos de Na y Nb con los de N:

$$\{Ng, Nr, Nh\} \cup \{Nr, Nh\} \cup \{Ne, Nf\} = \{Ng, Nr, Nh, Ne, Nf\}$$

$$\{Ng, Nr, Nh\} \cup \{Nr, Nh\} \cup \{Nk, Nh\} = \{Ng, Nr, Nh, Nk\}$$

$$\{Ng, Nr, Nh\} \cup \{Np, Nq\} \cup \{Ne, Nf\} = \{Ng, Nr, Nh, Np, Nq, Ne, Nf\}$$

$$\{Ng, Nr, Nh\} \cup \{Np, Nq\} \cup \{Nk, Nh\} = \{Ng, Nr, Nh, Np, Nq, Nk\}$$

- (2) Se suprimen de la etiqueta generada los entornos inconsistentes y se aplican criterios de simplificación tales como supresión de redundancias y supresión de entornos subsumidos en otros, ya que si un nodo puede deducirse de un entorno también puede obtenerse de

todos sus superconjuntos. Al aplicar criterios de simplificación sobre los cuatro entornos obtenidos antes, la etiqueta final de N se quedaría sólo con dos de ellos:  $\{Ng, Nr, Nh, Ne, Nf\}$  y  $\{Ng, Nr, Nh, Nk\}$ ; porque el tercero es un superconjunto del primero y el cuarto lo es del segundo.

- (3) Si la etiqueta obtenida es la misma que la de partida el proceso finaliza. Si no es igual pero es contradictoria, es decir, en alguno de sus entornos aparecen dos hipótesis incompatibles, entonces se computa la etiqueta del correspondiente nodo *no-good* y se eliminan de todas las etiquetas del retículo aquellas que contengan alguno de los entornos de la etiqueta *no-good*. De esta manera se provoca un efecto de no monotonía derivado de la modificación de las etiquetas de los nodos ya que el conjunto de creencias del sistema, o conjunto de afirmaciones justificadas que éste puede hacer, cambia.

En otro caso, se aplica el mismo proceso recursivamente a todos los nodos que incluyen al de partida en sus justificaciones utilizando en ello los nuevos contextos del nodo de partida.

Un aspecto a considerar, es la interpretación de las etiquetas lingüísticas utilizadas en la formulación de las preguntas. Por ejemplo, una pregunta del tipo *¿qué pasaría si H1 o H2?* hace que se añadan al retículo al menos dos justificaciones, una apoyada por H1 y otra por H2; así como la pregunta formulada como *¿qué pasaría si H1 y H2?* hace que se genere al menos una justificada por las dos hipótesis simultáneamente.

Por otra parte, dar soporte a secuencias de preguntas como las descritas al inicio de esta sección (*¿qué pasaría si H1?* seguido de *¿y si también sucediese H2?*, o *¿qué pasaría si H1?* seguido de *¿qué pasaría si H2?*) sólo requiere poder conocer cuál ha sido la última pregunta planteada por el usuario y bajo qué hipótesis, ya que sobre la gestión del retículo de creencias no tiene mayores consecuencias salvo respecto al número de justificaciones que recibe el ATMS y el tiempo en el que las recibe.

### 6.5.2.2 Acceso a información

Uno de los aspectos más útiles del uso del modelo ATMS para gestionar la memoria de un sistema es que proporciona rápidamente todos los conceptos



que son deducibles del estado de creencias del sistema, sirviendo como un depósito de información fácilmente accesible para los métodos de resolución de problemas.

Cuando el especialista en la realización de una tarea necesita decidir cómo llevarla a cabo, es decir, decidir el método que ha de aplicarse, y para ello uno de los aspectos a considerar es la disponibilidad de determinada información en la memoria, lo que hace es preguntar si existe algún valor para un determinado par *<objeto, atributo>*. En esta situación pueden darse varios casos:

- ⇒ Que exista y sea único, en cuyo caso el proceso de elección del método continúa en la forma especificada en el conocimiento de selección de métodos del especialista. Que exista más de uno, en cuyo caso es necesario aplicar criterios que inclinen la selección en uno u otro sentido. Dichos criterios van a estar basados en la forma de las preguntas y en la antigüedad de dichos valores. Por ejemplo, del análisis de una pregunta se puede concluir que hay que usar el valor obtenido a partir de datos reales, es decir, el obtenido a partir de premisas, o bien que hay que utilizar el que se dedujo como resultado de un razonamiento basado en una serie de hipótesis. Por otra parte, si la elección se inclina por un valor de este último tipo y se tienen varios valores obtenidos de razonamientos hipotéticos para el mismo par *<objeto, atributo>*, el criterio a aplicar podría ser tomar aquél deducido más recientemente, al entender que éste se aproxima más al modo en el que transcurren las conversaciones entre personas.

Por la forma que pueden tener las consultas de un usuario, podría ser que la búsqueda del nodo estuviese sujeta a condiciones, es decir, que fuese necesario que hubiese sido deducido a partir de determinadas hipótesis planteadas en la pregunta. En este caso, el mecanismo del ATMS proporcionaría rápidamente qué información está justificada por dichas hipótesis y bastaría con mirar si en ella se incluye algún valor para ese par *<objeto, atributo>*; o lo que es lo mismo, se seleccionarían en principio todos los nodos cuya etiqueta contenga algunos de los entornos de esas hipótesis, filtrando aquellos en los que coincide el par *<objeto, atributo>* con el de partida.

- ⇒ Que no exista, en cuyo caso el especialista sabrá que la resolución de su problema pasa por la realización de una tarea previa de la que se extraiga la información que necesita, por lo que el método elegido lo primero que hará será invocar al especialista en esa tarea para después generar la información solicitada.

### 6.5.2.3 Revisión de información

Todas las conclusiones anteriores se han obtenido suponiendo que las hipótesis empleadas en la formulación de las distintas preguntas de un diálogo y los resultados alcanzados por los diferentes procesos de razonamiento tras asumirlas, todos ellos contenidos en el conjunto de creencias del sistema, no presentan incompatibilidades. En el momento en el que se detecte algún tipo de incompatibilidad es necesario que el mecanismo de mantenimiento de la verdad del ATMS revise el conjunto de creencias hasta conseguir restaurar la consistencia del retículo de datos. Estas incompatibilidades pueden detectarse o bien porque están declaradas como tales en una base de conocimiento de *no-goods*, o bien porque en el transcurso de un diálogo el usuario se da cuenta de que algunas de las conclusiones obtenidas previamente deben ser revisadas porque se basan en hipótesis que en realidad son contradictorias, es decir, que no pueden darse simultáneamente.

Esto significa que las incompatibilidades entre hipótesis pueden observarse antes de llevar a cabo cualquier razonamiento, si el sistema conoce a priori las combinaciones de hipótesis *no-good* del dominio problema, puesto que lo primero que hace el razonador metanivel antes de generar la respuesta a una pregunta del usuario es consultar al ATMS sobre la consistencia del conjunto de hipótesis sobre las que ha de razonar; o después de haber razonado en base a ellas, si es el usuario el que declara la inconsistencia. En el primer caso el sistema no modificaría el estado de creencias, notificaría al usuario la incompatibilidad y esperaría a que éste le plantease una nueva pregunta, mientras que en el segundo el sistema le proporcionaría al ATMS una justificación de un nodo *no-good* apoyada por las hipótesis incompatibles y éste actuaría en consecuencia en la forma descrita anteriormente.

Por ejemplo, partiendo del estado de creencias descrito por la tabla de la figura 6.12, supóngase que aunque el sistema en alguna etapa previa del diálogo haya concluido que d8 es cierto basándose en las hipótesis h3 y h6 y en la



información deducida d7, el usuario se da cuenta de que no es razonable la coexistencia de h3 y d7 y lo declara como un *no-good*. Ante esta información el sistema debe revisar todas las creencias que tenga apoyadas en conjuntos de hipótesis que incluyan a h3 y a alguno de los contextos de h7, eliminando todos los contextos que se encuentren en esa situación. Así, en este ejemplo, el estado del conjunto de creencias una vez hecha la revisión sería el que muestra la figura 6.13, en la que puede observarse el hecho de que ya no puede deducirse d8 si se dan por válidas las hipótesis h3 y h6, cuando en el estado de conocimiento previo al *no-good* ésto sí era cierto. El mismo efecto se habría producido si el sistema hubiese dispuesto a priori de la información sobre el *no-good*, con la diferencia de que en este caso el ATMS habría podido prevenir al resolvidor de problemas sobre la inconsistencia del conjunto de hipótesis sobre el que estaba dispuesto a llevar a cabo un razonamiento.

Nodo	Descripción	Etiqueta	Justificaciones
n1	<obj1, atr1, v1>	{ }	(n1)
n2	<obj2, atr2, v2>	{ }	(n2)
h3	<obj3, atr3, v3>	{h3}	(h3)
h4	<obj4, atr4, v4>	{h4}	(h4)
h5	<obj5, atr5, v5>	{h5}	(h5)
h6	<obj6, atr6, v6>	{h6}	(h6)
d7	<obj7, atr7, v7>	{h4, h5} {h6}	(n1,h4,h5) (n2,h6)
d8	<obj8, atr8, v8>	{h4,h5} {h5,h6}	(h5,d7) (h3,h6,d7)
no-good	<no-good, si>	{h3,h4,h5} {h3,h6}	(h3,d7)

Figura 6.13: Representación del estado de creencias de un sistema tras incorporar un no-good

6.5.2.4 Eliminación de información

Un problema asociado al uso del modelo ATMS para gestionar la memoria de una interacción es la explosión combinatoria provocada durante la actividad de generación de nuevas etiquetas y el crecimiento desmesurado del retículo de nodos o creencias, sobre todo en el caso de sistemas complejos con posibilidades amplias de diálogo entre los usuarios y el sistema ya que en estos



casos el resolutor de problemas genera mucha información y por tanto, tras la ejecución de un proceso de inferencia, aporta numerosas justificaciones.

Por ello es necesario introducir algún mecanismo de olvido o limpieza de la memoria para que pueda seguir añadiéndose nueva información, de forma análoga a como sucede en las personas. Siguiendo con este paradigma, la solución adoptada pasa por articular lo que sería la memoria a corto plazo de las personas, prescindiendo de la capacidad de memorizar a largo plazo. El restringir el tamaño de la memoria a las creencias adoptadas recientemente no tiene demasiado coste en el contexto de sistemas de ayuda a la decisión en tiempo real ya que estos sistemas han de reconsiderar todas sus conclusiones cada vez que reciben nuevos volúmenes de información procedente de la infraestructura telemática de soporte.

De acuerdo con estas consideraciones, el tipo de medidas adoptadas ha sido el siguiente:

- Definir un número máximo de instancias para cada par *<objeto, atributo>* de manera que una vez alcanzado ese número la incorporación de un nuevo valor para ese par hace que se elimine el más antiguo.
- Introducir en el retículo justificaciones sólo de los conceptos que son relevantes para la interacción, es decir:
  - (i) aquellos que pueden influir en la selección de métodos llevada a cabo por el razonador metanivel,
  - (ii) los que puede utilizar el usuario para formular preguntas en forma de hipótesis, y
  - (iii) los que se utilizan para definir justificaciones *no-good*.

## 6.6 Conclusiones

En este capítulo se ha descrito una arquitectura general, llamada ROHCI, orientada al diseño de sistemas inteligentes caracterizados por su capacidad



para soportar un modelo avanzado de interacción con los usuarios, definido en los términos descritos en el capítulo 5, articulado en torno a un diálogo usuario-sistema.

La definición de esta arquitectura es el resultado de un proceso iniciado con el análisis de las características del modelo de interacción objetivo, tras el que se establecieron las necesidades de conocimiento de un sistema capaz de soportar dicho modelo. Para poder alcanzar dichos objetivos de interacción el sistema debía disponer de cuatro clases de conocimiento:

- Conocimiento sobre cómo resolver problemas en el dominio de definición del sistema.
- Conocimiento sobre los términos en los que se puede desarrollar un diálogo, su estado actual y su evolución reciente.
- Conocimiento sobre el tipo de usuarios con los que puede interactuar.
- Conocimiento sobre sus propias capacidades de resolver problemas y responder preguntas.

La arquitectura ROHCI maneja estos cuatro cuerpos de conocimiento, en la dirección marcada por los objetivos de interacción, apoyándose en los siguientes elementos:

- ◊ Un modelo de interacción estructurado por tipos de preguntas que responden a las necesidades de información de los usuarios y definen los términos en los que pueden desarrollarse los diálogos.
- ◊ Un Espacio de Resolvedores de Problemas concebido como un modelo de conocimiento estructurado jerárquicamente donde reside el conocimiento de resolución de los problemas propios del dominio de la aplicación y las piezas software capaces de llevar a cabo el razonamiento.
- ◊ Un modelo de razonamiento metanivel, donde reside la capacidad reflexiva del sistema, que navega en el medio de resolución de problemas para proponer, en función de las características y preferencias del usuario y del contexto de la interacción, el tipo de respuestas que deben proporcionarse al usuario y el modo de obtenerlas. Este módulo reflexivo se apoya en dos tipos de conocimiento:

- Conocimiento sobre escenarios de interacción con los diferentes tipos de usuarios en los que se representan los criterios que deben guiar la búsqueda de la respuesta esperada por el usuario en función de las etapas previas del diálogo y de las intenciones comunicativas y características del usuario. Estos criterios se definen como atributos de interacción y describen características de los métodos de resolución de problemas deseables desde la perspectiva de la interacción.
  - Conocimiento especializado en el diseño de estructuras de razonamiento cuyos elementos son métodos de resolución de problemas elegidos en base a los atributos de interacción decididos anteriormente y el estado de la interacción.
- ◊ Una memoria de la interacción gestionada por un sistema de mantenimiento de la verdad, de tipo ATMS, que asegura la coherencia de la interacción usuario-sistema en el tiempo.

La arquitectura ROHCI, descrita en este capítulo, ofrece una solución genérica de diseño de sistemas inteligentes con amplias capacidades interactivas que es aplicable en diferentes dominios problema y que daría lugar a aplicaciones reutilizables, flexibles y escalables. Estas tres importantes características están soportadas por los siguientes elementos:

- ⇒ *Reusabilidad*. Se sustenta en la modularidad y estructuración del modelo de conocimiento. Una aplicación construida sobre ROHCI proporciona una solución a una clase de problemas que está fundamentalmente contenida en el Espacio de Resolvedores de Problemas (ERP). La organización jerárquica de los componentes del ERP hace que los niveles superiores de la estructura recojan aspectos más relacionados con la clase de problemas que se resuelven, mientras que a medida que se desciende en la jerarquía, los componentes expresan formas de resolver problemas más orientadas al dominio concreto de la aplicación. Así, esta estructura es reutilizable a distintos niveles en diferentes dominios dependiendo de las similitudes entre el dominio problema original, y el tipo de problemas a resolver y los métodos utilizados para resolverlos en el dominio problema destino. En general, el modelo de resolución de problemas a metanivel será



reutilizable en la misma medida en que lo sea el de nivel objeto. Por otra parte, el grado de reutilización del modelo de escenarios de interacción ha de evaluarse de forma separada al de los dos anteriores, ya que sólo depende de su capacidad para reflejar las peculiaridades propias de los usuarios del dominio problema destino con respecto a su tipología, intenciones o nivel de experiencia.

- ⇒ *Flexibilidad.* Este aspecto está soportado por la capacidad del sistema de adaptar su comportamiento en la resolución de problemas a las características de la interacción que se esté desarrollando. Así, la flexibilidad del sistema se obtiene a dos niveles: (i) respecto al modo de resolver problemas y, (ii) respecto al tipo de usuarios con los que el sistema puede interactuar.
- ⇒ *Escalabilidad.* Se sustenta en la modularidad y diferenciación entre tipos de conocimiento utilizados. Normalmente, la ampliación de las funcionalidades de un sistema construido sobre el modelo ROHCI afectará, en mayor o menor medida, a tres componentes: (i) el modelo de interacción, es decir, la especificación del conjunto de preguntas con las que usuarios y sistema se comunicarán, (ii) el modelo de escenarios de interacción, es decir, el conocimiento que tiene el sistema de las características y objetivos comunicativos de los usuarios del dominio problema en diferentes escenarios de operación, y (iii) el modelo de resolución de problemas. En general, la incorporación de nuevas preguntas irá acompañada de la especificación de nuevas formas de resolver problemas, cuyo aprovechamiento se gestiona atendiendo a criterios proporcionados por el modelo de escenarios de interacción. Sin embargo, podrían introducirse modificaciones en alguno de estos componentes sin que los demás se viesen afectados en la misma medida. Por ejemplo, una redistribución de responsabilidades y roles en un entorno de operación supone una modificación del modelo de escenarios de interacción que no tiene por qué afectar a los otros dos componentes.

La arquitectura ROHCI muestra cómo un sistema basado en el conocimiento puede adaptar su comportamiento a las necesidades de su diálogo con un usuario, mediante la integración de diferentes técnicas de Inteligencia Artificial y elementos propios de los modernos interfaces de usuario.

## **Parte III: APLICACIÓN**





## **7. Gestión de Transporte Público en Tiempo Real**

---

Este capítulo presenta la aplicación de la arquitectura ROHCI en el diseño de un sistema de gestión de transporte público en tiempo real para el proyecto FLUIDS, financiado por la Comisión Europea dentro del programa de Aplicaciones Telemáticas en el Cuarto Programa Marco. Esta aplicación tiene dos perspectivas: (i) por una parte, proporciona una solución al problema de gestionar la red de autobuses urbanos de la ciudad de Turín, y por otra (ii) ofrece un modelo avanzado de interacción usuario-sistema.

En primer lugar, se describe cómo las peculiaridades de los escenarios de toma de decisiones en tiempo real condicionan las características de los sistemas informáticos contruidos para dar soporte a los operadores en tales contextos. En segundo lugar, se describe el problema de la gestión del transporte público en la ciudad de Turín, las situaciones conflictivas que se suelen presentar y el tipo de acciones que se toman para solventarlas. Después, se presenta el modelo de diálogo que desea establecerse entre el sistema y los distintos tipos de usuarios que existen en este dominio. A continuación, una vez establecido el tipo de interacción objetivo, se describen cómo son los componentes del sistema final que da soporte a ese tipo de interacción: el Espacio de Resolvedores de Problemas y el Razonador Metanivel. Finalmente, se describe el prototipo desarrollado y se hace una evaluación de sus ventajas e inconvenientes frente a la solución existente en el centro de control de Turín.

### **7.1 Sistemas de Ayuda a la Decisión en Tiempo Real**

El desarrollo de las telecomunicaciones ha provocado un cambio drástico en el modo de controlar sistemas físicos complejos. Tradicionalmente, los responsables de mantener un nivel adecuado de funcionamiento en estos



sistemas se apoyaban en información escasa obtenida con procedimientos casi manuales y en impresiones visuales del estado del sistema. La implantación de nuevas tecnologías de sensorización y comunicaciones ha obligado a estas personas a desenvolverse en un nuevo entorno de trabajo caracterizado por la recepción de cantidades ingentes de información en cortos intervalos de tiempo, que ha de usarse para tomar decisiones sobre acciones en los elementos de control del sistema.

Ante este escenario se plantea la necesidad de disponer de sistemas automáticos que faciliten el trabajo, obteniendo el máximo rendimiento de la información recibida y ofreciendo al operador respuestas que permitan una reacción a tiempo frente a una situación problemática, impidiendo el empeoramiento de la misma o minimizando sus consecuencias. Es en este sentido de llegar *a tiempo* en el que se conciben los sistemas de ayuda a la decisión en tiempo real considerados en este volumen, en contra de la interpretación tradicional de tiempo real como generación de respuestas bajo grandes restricciones de tiempo.

La caracterización de un sistema físico, de cara a su gestión, supone definir los siguientes elementos, de acuerdo con [Cuenca, 89]:

- Un conjunto de acciones externas que pueden ser producto del entorno en el que opera el sistema o que pueden haberse generado como consecuencia de decisiones previas tomadas por el responsable de la gestión.
- Un conjunto de componentes que definen la estructura física del sistema.
- El comportamiento del sistema descrito en función de los cambios de estado de sus componentes y las interacciones entre los mismos.
- Algún criterio de evaluación de la calidad de funcionamiento del sistema para poder valorar la distancia con respecto a los objetivos de gestión predefinidos y tomar las medidas de ajuste necesarias.

Entonces, un sistema capaz de realizar una gestión efectiva de un sistema físico descrito en los términos anteriores debería ser capaz de llevar a cabo las siguientes actividades:

- Predecir la evolución de las acciones externas al sistema.
- Interpretar el efecto, es decir el cambio de estado, que provocan las acciones externas sobre el estado del sistema.
- Identificar los estados problemáticos actuales y los previstos a corto plazo.
- Generar propuestas de cambios en la estrategia de control vigente que ayuden a mejorar o prevenir los estados problemáticos.

A la vista de estas actividades pueden identificarse un conjunto de tareas genéricas que emplean conocimiento especializado para:

- Predecir la evolución a corto plazo de las acciones externas.
- Simular el comportamiento del sistema ante diferentes escenarios de acciones externas.
- Clasificar los tipos de problemas que pueden presentarse en los distintos escenarios de comportamiento del sistema.
- Decidir qué acciones han de tomarse para actuar sobre las causas de los problemas o sobre sus efectos.

Un sistema capaz de llevar a cabo estas tareas debería ser capaz entonces de soportar un modelo de interacción que diese respuesta a tres grandes clases de preguntas:

- *Qué está pasando.* Orientadas a obtener una descripción del estado en el que se encuentra el sistema bajo control.
- *Qué podría pasar.* Dirigidas a evaluar la clase de situación que podría presentarse a corto plazo en este sistema bajo ciertas condiciones del entorno o como consecuencia de determinadas acciones externas.
- *Qué debería hacerse.* Dirigidas a obtener planes de acciones de control que resuelvan problemas detectados o previstos, o que minimicen sus efectos



sobre el estado del área. Esta clase de preguntas también podría formularse bajo distintas hipótesis de comportamiento o evolución del área problema.

Un sistema inteligente capaz de llevar a cabo las tareas anteriores debe actuar en conjunción con el usuario responsable de tomar las decisiones pero siempre asumiendo en esta comunicación el rol de un asistente que actúa bajo demanda. La eficacia de la ayuda prestada por este tipo de sistemas pasa por la capacidad de los mismos para transmitir al usuario sus conclusiones en forma comprensible y justificada, de manera que el usuario pueda asumir las decisiones sugeridas por el sistema como propias. Para ello el diseño de estos sistemas debe contemplar la definición de un modelo de comunicación usuario-sistema con el que el usuario pueda extraer toda la información que necesita para asegurarse de la verosimilitud de las propuestas del sistema mediante sesiones de preguntas y respuestas.

## **7.2 El Problema de la Gestión del Transporte Público**

La gestión de una red de transporte público en una gran ciudad es un problema complejo que, hoy por hoy, todavía está fundamentalmente soportado por operadores humanos. En la mayoría de los casos, la supervisión del funcionamiento del servicio previsto y la resolución de los problemas generados en la red se lleva a cabo mediante comunicaciones directas vía radio entre los conductores de los vehículos en servicio y los operadores localizados en el centro de control.

Una solución de gestión más avanzada es la que se aplica en la ciudad de Turín, la cual dispone de una infraestructura telemática consistente en una conexión bidireccional entre un sistema de información localizado en el centro de control, llamado S.I.S., y los vehículos que circulan por la red. Esta conexión se utiliza para intercambiar datos y comunicaciones orales con el fin de centralizar la información en tiempo real sobre el estado del servicio de los vehículos que circulan por la red con respecto al plan previsto. El escenario desde el que se lleva a cabo esta gestión es el centro de control de la empresa municipal de transporte de Turín, Transporte Torinesi, en el que un número aproximado de 6 a 10 operadores, dependiendo del tipo de día y de la franja

horaria, controlan simultáneamente la evolución del servicio de 4 a 6 líneas cada uno de ellos, supervisados por un jefe de sala responsable de distribuir los recursos de acuerdo con las necesidades de las líneas y de dar la autorización a acciones de control que impliquen a varias líneas.

El S.I.S. determina automáticamente la localización de todos los vehículos en servicio cada minuto a partir de la información periódica recibida sobre la distancia recorrida por cada uno de ellos y el reconocimiento de la llegada a una parada, mediante sensores en las ruedas y las puertas respectivamente; a la vez que calcula las diferencias entre la marcha de cada uno de los vehículos con respecto a su horario previsto. Esta información se presenta a los operadores mediante un interfaz gráfico en el que sobre una representación de la línea se muestra, para cada vehículo, su posición y un código de color con el que se indica el grado de retraso acumulado por ese vehículo.

La observación de discrepancias entre la evolución prevista para cada vehículo y la observada a partir de los datos enviados desde la calle, o la aparición de eventos inesperados como averías en los vehículos u obstáculos en las líneas provoca la aplicación de estrategias de actuación a dos niveles:

- El *microcontrol*, utilizado en el caso de retrasos poco significativos y realizado de forma automática, es decir, sin intervención del operador. Consiste en proponer a los conductores ajustes sencillos sobre su recorrido, para lo cual los vehículos están equipados con un tablero electrónico que muestra una gama de diferentes colores e iconos en función del tipo de problema detectado y ante los que los conductores saben las medidas que deben adoptar.
- El *macrocontrol*, aplicado cuando la desviación observada es importante o se han producido eventos inesperados. En este caso, se requiere la intervención directa del operador para que éste determine las acciones de control necesarias para restaurar el servicio planificado y las comunique a los conductores afectados vía radio. En el caso de que las decisiones adoptadas supongan hacer uso de los recursos de control disponibles en las líneas, como vehículos de reserva, los operadores necesitan la autorización del responsable de sala.



Aunque el sistema S.I.S constituye una herramienta de gran ayuda para la gestión del transporte público en Turín al centralizar toda la información sobre el estado de las diferentes líneas, no explota todo el potencial de la información que maneja ya que para ello debería de proveer de algún tipo de facilidad orientada a resolver los problemas derivados del hecho de que el proceso de toma de decisiones está todavía completamente soportado por operadores. En consecuencia, se observan los siguientes problemas:

- Los operadores deben supervisar simultáneamente varias líneas, de las cuales reciben una gran cantidad de información ofrecida a muy bajo nivel de detalle que deben interpretar para poder detectar la presencia de problemas y determinar sus causas.
- Los operadores no reciben del sistema ninguna sugerencia sobre el tipo de acciones de control que serían más adecuadas para resolver los problemas observados en las líneas, por lo que éstos tienen que dedicar parte de su tiempo a realizar cálculos rutinarios sobre la disposición que deberían tener los vehículos y cómo llegar a ella.
- El responsable de la sala de control no dispone de ninguna facilidad por parte del sistema para obtener una visión global del estado de la red o de alguna parte de la misma en la que confluyen varias líneas, por lo que para decidir las estrategias de control a adoptar en cada momento ha de comunicarse directamente con cada uno de los operadores implicados en la gestión de las líneas con problemas.

La construcción de un sistema de ayuda a la decisión basado en el conocimiento que soporte un modelo de interacción como el que propone esta tesis, en principio, reportaría como ventajas:

- Una mejora de la calidad de la operación al procesar la información a un nivel superior de abstracción, ofreciendo así visiones más estratégicas de las líneas adaptadas al tipo de usuario que las solicita.
- Una disminución del esfuerzo requerido para supervisar el funcionamiento de varias líneas, liberando así a los operadores de labores rutinarias y aumentando su productividad, así como mejorando la

eficiencia del responsable de sala al poder recabar la información que necesita para tomar decisiones sin molestar a los operadores.

Por tanto, este sistema actuaría como un experto en control de transporte público con capacidad para:

- (i) filtrar la información más relevante para el operador,
- (ii) centrar la atención del operador sobre problemas potenciales,
- (iii) sugerir y justificar acciones de control que resolverían los problemas existentes, y
- (iv) estudiar la viabilidad de alternativas propuestas por el operador.

El acceso a estas funcionalidades se canalizaría a través de un modelo de interacción entre el sistema y sus diferentes tipos de usuarios, formalizado en los términos descritos en el capítulo 3, que ilustra las posibilidades del concepto de interacción inteligente presentado en esta tesis.

### 7.2.1 Terminología de gestión de transporte público

Una red de transporte público está formada por un conjunto de líneas, cada una de las cuales descrita con una secuencia de paradas y *puntos de cambio* agrupados en varias secciones unidireccionales de línea. Asociada a una línea hay también dos direcciones de recorrido definidas como un par ordenado de las dos paradas identificadas como cabecera y fin de línea. Los puntos de cambio se corresponden con la posición de semáforos y se emplean, o bien para definir puntos de relevo de conductores o bien como referencia para definir puntos en los que es posible cambiar la dirección de un vehículo.

Las líneas pueden controlarse atendiendo a dos criterios: (i) por *horario*, y (ii) por *frecuencia*. En el primer caso, el interés se centra en respetar la puntualidad de los pasajeros, mientras que en el segundo el objetivo es mantener la frecuencia de autobuses programada. El tipo de gestión aplicado sobre una línea está predefinido en función de la importancia de la línea. Las líneas menos importantes suelen regularse siempre por horario, mientras que las más relevantes se controlan por frecuencia durante la mayor parte del día para



pasar a ser reguladas por horario cuando las horas de mayor demanda han pasado.

Los vehículos que dan servicio a la red de transporte público se clasifican en tres grupos:

- (i) *Vehículos de servicio regular*, que se incorporan a la línea desde los depósitos y cuyo número varía en función de la demanda prevista.
- (ii) *Vehículos de reserva*, que se emplean para resolver problemas de interrupciones temporales del servicio en cualquiera de las líneas y que se localizan, en número fijo, en áreas específicas distribuidas por la ciudad denominadas *áreas de reserva* en las que nunca hay disponible más de un vehículo de reserva.
- (iii) *Vehículos de reparaciones*, localizados en algunas de las áreas de reserva, que se envían a las líneas cuando alguno de los vehículos de servicio sufre una avería.

## 7.2.2 Clases de problemas

La caracterización del estado de una red de transporte público puede realizarse a dos niveles de abstracción: (i) atendiendo al estado de los vehículos que circulan por la línea o (ii) atendiendo al estado de la propia línea. En el primer caso pueden distinguirse los siguientes problemas:

- Discrepancias entre la evolución del servicio previsto y el observado para algunos vehículos, a la vista de los datos enviados por el sistema de información, provocadas por unas malas condiciones de tráfico. Estas discrepancias pueden suponer tanto un adelanto como un retraso respecto al horario previsto.
- Bloqueo de vehículos debido a un corte en alguna sección de la línea que les impide seguir la ruta prefijada.
- Averías detectadas en los vehículos. Estas averías se clasifican en tres niveles de gravedad: (i) aquellas que pueden resolverse con un vehículo de reparaciones en menos de 5 minutos, (ii) aquellas cuya reparación

requiere más de 5 minutos pero que no impiden al vehículo llegar hasta la parada final de la línea y, (iii) aquellas que dejan inmovilizado el vehículo en medio de la línea.

Por otra parte, la calificación del estado de una línea está directamente relacionada con el estado de los vehículos que circulan por ella, por lo que como estados problemáticos de una línea cabe considerar los siguientes:

- Presencia de paradas bloqueadas que resultan inaccesibles a los vehículos de servicio debido a obstáculos temporales, como accidentes, o permanentes, como obras en la calzada.
- Presencia de retrasos en gran parte de los vehículos de servicio distribuidos uniformemente a lo largo de la línea.
- Presencia de retrasos en vehículos puntuales localizados en puntos dispersos de la línea.

### 7.2.3 Acciones de control

A la vista de los problemas identificados anteriormente pueden llevarse a cabo acciones de control a distintos niveles de complejidad orientadas a restaurar el servicio:

⇒ Acciones adecuadas para resolver problemas de ligeros retrasos y que se resuelven con pequeñas modificaciones del servicio previsto. Aquí se incluyen:

- *Salto de algunas paradas*, siempre que la demanda de los pasajeros lo permita, es decir, que no haya pasajeros esperando en dichas paradas ni ninguno de los pasajeros que van en el vehículo desee bajarse;
- *Desvíos por rutas alternativas* para superar las paradas bloqueadas en la trayectoria de una línea. Podría considerarse similar al anterior, pero en este caso el recorrido transcurre fuera de la línea;
- *Reducción del tiempo de descanso* previsto en las cabeceras de línea.



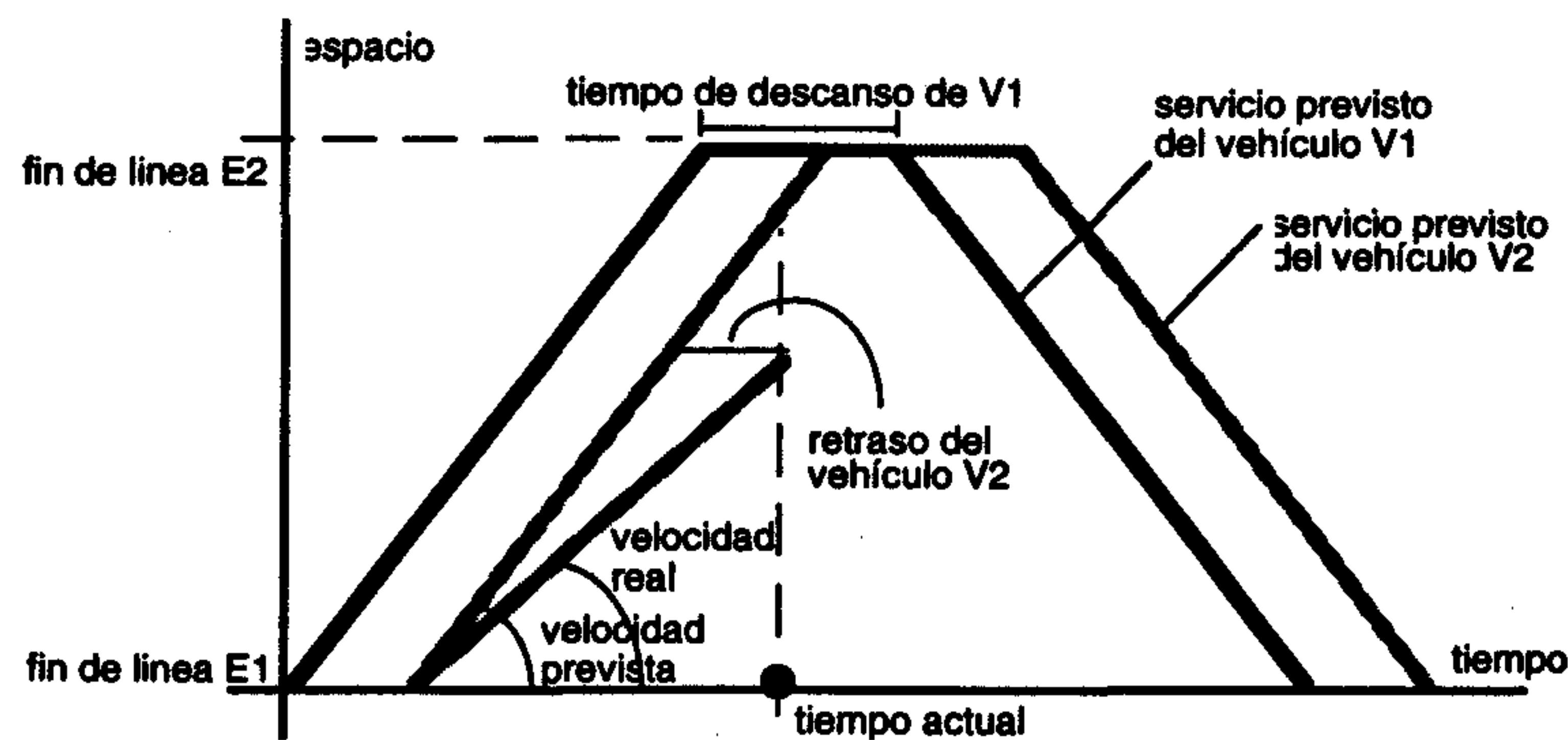


Figura 7.1: Representación de la evolución observada y prevista de los vehículos

⇒ Acciones complejas necesarias para restaurar el servicio ante retrasos importantes que requieren alterar significativamente la evolución de los vehículos implicados para que éstos recuperen la posición prevista. Este tipo de acciones requiere una detallada planificación por parte de los operadores ya que implican a varios vehículos y han de realizarse en varias etapas. La descripción del impacto de estas acciones sobre el servicio previsto para un vehículo puede representarse gráficamente como se muestra en la figura 7.1.

Hay dos clases de acciones complejas:

- **Restricción del servicio.** Se aplica cuando dos vehículos están muy retrasados y muy próximos, y no hay disponibles vehículos de reserva. Entonces, se transfieren los pasajeros de uno al otro y el que queda vacío cambia de dirección, posicionándose aproximadamente sobre la parada en la que debería encontrarse si no hubiese sufrido retraso. Su representación gráfica es la que muestra la figura 7.2.

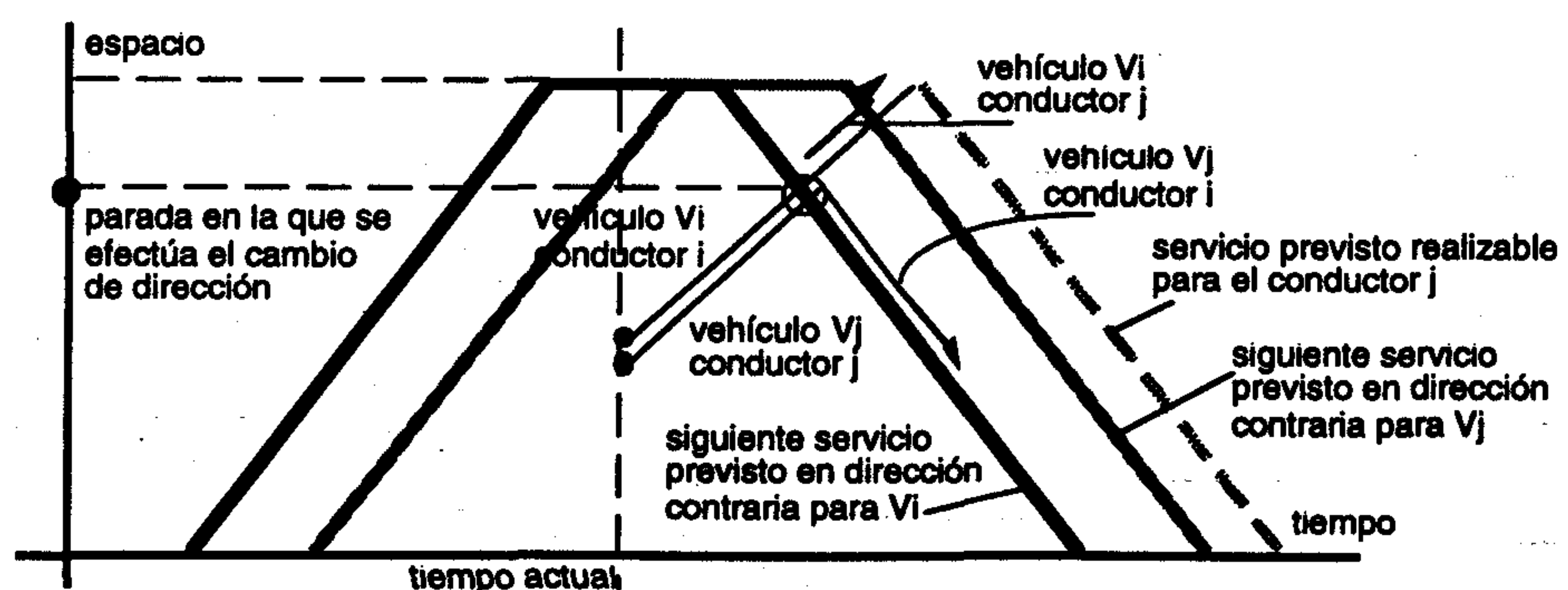


Figura 7.2: Aplicación de una acción de restricción de servicio

- **Procedimiento Spola.** Si se detecta algún vehículo muy retrasado y hay disponibles vehículos de reserva, entonces se envía uno de ellos a cubrir el tramo que queda desatendido por el vehículo retrasado en la dirección contraria. Para configurar esta decisión es necesario tener en cuenta el horario previsto de este vehículo, para poder determinar el momento en el que el vehículo de reserva debe iniciar el recorrido de la línea en la dirección contraria a la del vehículo retrasado; y el tiempo que requerirían los vehículos de reserva elegibles para alcanzar la cabecera de la línea a la que se dirige el vehículo retrasado. Además, se determina un punto o área de cruce entre los dos vehículos que circulan en direcciones contrarias, el de servicio y el de reserva, para que los conductores se cambien de vehículo. Así, el conductor del vehículo retrasado pasa a conducir el vehículo de reserva recuperando el horario previsto y el conductor de reserva lleva el vehículo retrasado al final de línea y de allí al área de reserva correspondiente, convirtiéndose en vehículo de reserva. El efecto de este tipo de acción puede verse en la figura 7.3.

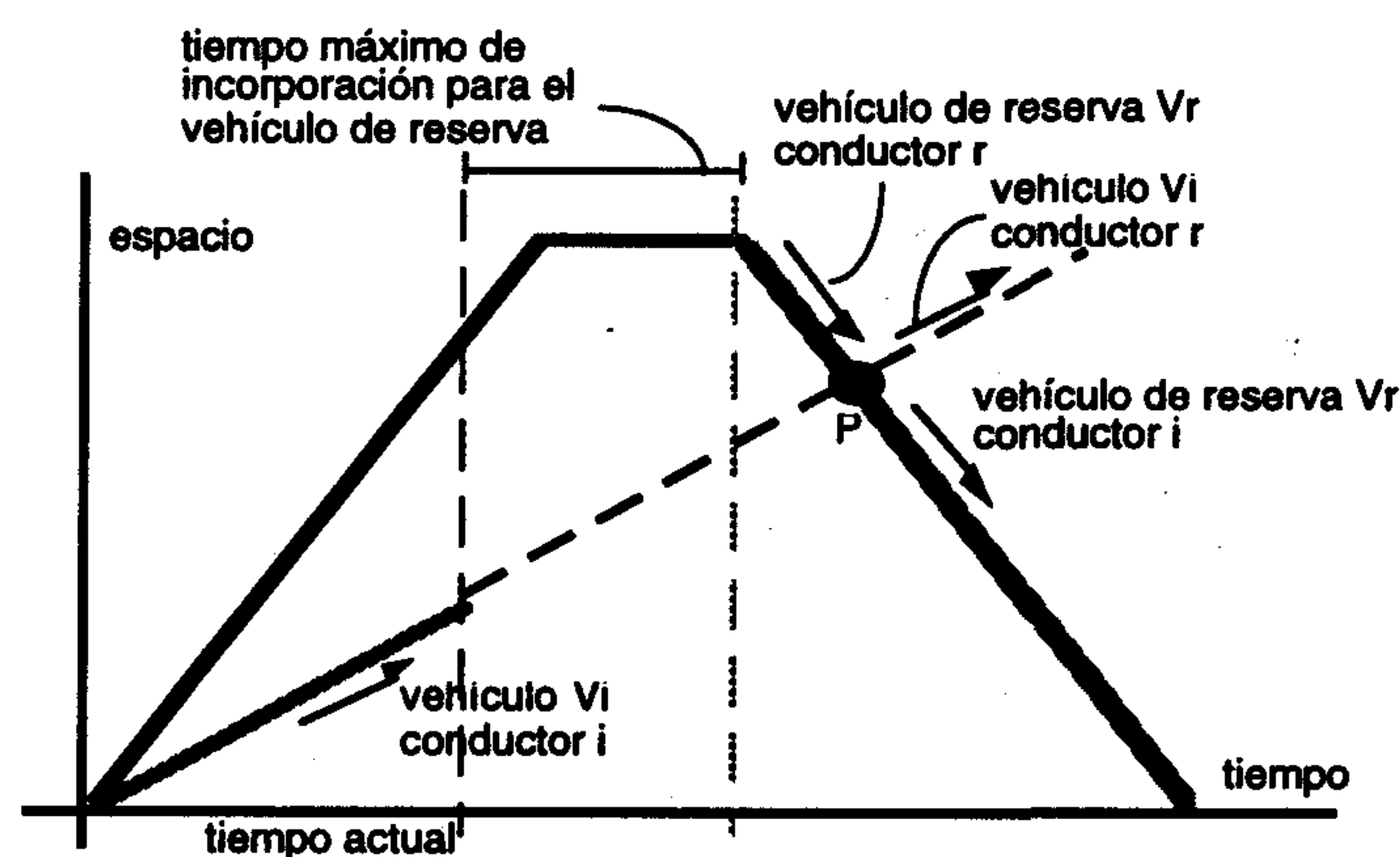


Figura 7.3: Aplicación de un procedimiento Spola

⇒ Envío de un vehículo de reparaciones para solventar averías detectadas en vehículos de servicio. Dependiendo de la gravedad de la avería esta acción puede llevarse a cabo de tres formas distintas:

- Si la avería parece reparable en menos de 5 minutos y no impide que el vehículo llegue al final de la línea, el vehículo de



reparaciones se envía al final de la línea a esperar al vehículo averiado y el tiempo de la reparación se compensa con una disminución del tiempo de descanso del vehículo en ese final de línea.

- Si la reparación de la avería requiere más de 5 minutos pero el vehículo no ha perdido la movilidad y puede alcanzar el final de la línea, entonces el vehículo de reparaciones se envía al final de la línea y para mantener la regularidad en el servicio se aplica un encadenamiento de cambios de conductores y vehículos, de manera que son sólo los conductores los que descansan el tiempo estipulado tras el que parten con el vehículo que acaba de llegar al final de la línea.
- Si la avería es grave e impide moverse al vehículo entonces el vehículo de reparaciones se envía al punto en el que se encuentre el vehículo averiado y, para recuperar el retraso generado por la avería se aplica un procedimiento Spola o bien uno de restricción del servicio, dependiendo de la disponibilidad de vehículos de reserva.

La clasificación de las acciones anteriores como acciones de microcontrol y macrocontrol puede verse en la tabla de la figura 7.4.

<b>Microcontrol</b>	reducción del tiempo de descanso
	salto de paradas
<b>Macrocontrol</b>	desvío por rutas alternativas
	restricción del servicio
	procedimiento Spola
	reparación rápida
	reparación al final de la línea
	reparación en medio de la línea

Figura 7.4: Clasificación de las acciones de control

## 7.3 El Modelo de Interacción Usuario-Sistema

De acuerdo con las características del dominio de transporte público, para cada uno de los tres tipos de preguntas - qué pasa, qué podría pasar, qué hacer -, las clases de preguntas que habría que considerar serían las siguientes:

### ⇒ **¿QUÉ ESTÁ PASANDO?**

- *¿qué está pasando en la línea <L>?*

Con esta pregunta se busca obtener una descripción o valoración del estado en el que se encuentra la línea L. Como los operadores supervisan simultáneamente el estado de varias líneas cabe considerar dos clases de respuestas: (i) una en la que se detallan los problemas encontrados, identificando las causas y los vehículos implicados y, (ii) otra expresada de forma abstracta que se limita a proporcionar una cualificación general del estado, por ejemplo empleando un código de colores. La segunda de ellas se usa en modo alerta para captar la atención del operador sólo cuando la situación de la línea lo requiera, permitiendo así que éste se centre en la gestión de líneas más problemáticas. Además, podría considerarse una tercera clase de respuesta en la que se incluyen llamadas de atención sobre eventos previstos a corto plazo que podrían incidir negativamente en la normal evolución del servicio en la línea.

- *¿qué está pasando con el vehículo <V> en la línea <L>?*

Ésta puede considerarse un zoom en un aspecto particular del estado de una línea, como es el comportamiento de uno de los vehículos que circulan por ella. En este sentido, la respuesta incluye información sobre posibles desajustes del servicio prestado por el vehículo con respecto al planificado, es decir, retrasos o adelantos respecto al horario, así como la presencia y tipo de averías detectadas en el vehículo.

### ⇒ **¿QUÉ PODRÍA PASAR?**

- *¿qué podría pasar en la línea <L>?*



Esta pregunta persigue una valoración del estado que puede alcanzar la línea L si se mantienen las condiciones actuales y no se aplica ninguna acción de control.

- *¿qué podría pasar en la línea <L> si el vehículo <V> es enviado al garaje?*

El objetivo es averiguar si, de acuerdo con las características de la línea y el entorno, los recursos disponibles en la línea son suficientes para gestionar la situación provocada por el envío de un vehículo al depósito.

- *¿qué podría pasar en la línea <L> si las condiciones del tráfico imponen una velocidad de <X> Km/h en el sector <S>?*

Esta pregunta interroga sobre el impacto que provocaría en el servicio un cambio en la velocidad media de un sector de la línea, es decir, si se produciría una interrupción importante del servicio o si éste podría continuar con normalidad.

- *¿qué podría pasar con el horario previsto del vehículo <V> de la línea <L> si las condiciones del tráfico imponen una velocidad de <X> Km/h en el sector <S>?*

Este es un caso particular de la pregunta anterior que se centra en las consecuencias que tal velocidad provocaría sobre un vehículo concreto, es decir, si haría que se produjese un retraso significativo o no.

### ⇒ **¿QUÉ DEBERÍA HACERSE?**

- *¿qué debería hacerse en la línea <L>?*

Con esta pregunta se busca obtener sugerencias de acciones a tomar, a la medida de los problemas observados, sobre la situación actual de la línea L. La respuesta puede ofrecer líneas generales de actuación, planes detallados de acciones a tomar, e incluso una valoración del impacto que se prevé con las acciones recomendadas.

- *¿qué debería hacerse en la línea <L> si no hay vehículos de reserva disponibles en las áreas <A1, A2,..., An>?*

Esta pregunta también podría formularse directamente a partir de la información sobre el estado de una línea y el de las áreas de reserva, o

bien a continuación de la pregunta general sobre lo que debería hacerse en L en aquellos casos en los que las acciones de control propuestas plantean hacer uso de determinados vehículos de reserva. En la respuesta podría confirmarse la estrategia de control propuesta previamente cambiando simplemente el área de reserva de la que tomar el vehículo, o bien podría configurarse un tipo diferente de acción de control.

- *¿qué debería hacerse en la línea <L> si las condiciones del tráfico imponen una velocidad de <X> Km/h en el sector <S>?*

Esta pregunta es similar a la primera pero en este caso el estado de línea sobre el que se solicitan las recomendaciones de acciones de control es el que se plantearía bajo la hipótesis de una determinada velocidad en uno de los sectores de la línea.

- *¿qué podría hacerse en la línea <L> con el vehículo <V> que está próximo a finalizar el servicio?*

Con esta pregunta se busca explorar la posibilidad de emplear vehículos de servicio que están a punto de volver al depósito para sustituir a vehículos averiados o hacer las funciones de un vehículo de reserva, debido al número limitado de vehículos de reserva disponibles para todas las líneas.

En cada uno de los tres grupos de preguntas se incluyen también las preguntas derivadas de las anteriores que solicitan la explicación de la respuesta ofrecida por el sistema. Así una pregunta *por qué?* después de *qué está pasando en la línea <L>?* busca la justificación del estado descrito por el sistema para la línea L, la misma pregunta *por qué?* planteada tras *qué debería hacerse en la línea <L>?* busca las razones que han llevado al sistema a proponer la acción o acciones de control descritas previamente, etc.

Además, también se incluyen dentro de estos grupos variaciones sobre las preguntas anteriores en las que se expresan conjunciones o disyunciones de hipótesis, como por ejemplo: *qué podría pasar en la línea <L> si el vehículo <V> es enviado al garaje o las condiciones del tráfico imponen una velocidad de <X> Km/h en el sector <S>?* y *qué debería hacerse en la línea <L> si no hay vehículos de reserva en las áreas <A1,...,Ak> y el vehículo <V> está próximo a finalizar el servicio?*



La tabla de la figura 7.5 presenta una enumeración de algunas de las clases de preguntas que pueden formularse y de la forma que tienen sus respuestas.

CLASES DE PREGUNTAS	CLASES DE RESPUESTAS
Qué está pasando	
¿Qué está pasando en la línea <L>?	Respuesta simple y abstracta que cualifica globalmente el estado de la línea.
	Estado global de la línea mas una valoración de lo que debe ser analizado desde la perspectiva del macrocontrol
	Una respuesta detallada con la localización y estatus de cada uno de los vehículos con respecto a su horario
¿Qué está pasando con el vehículo <V> en la línea <L>?	Valoración cualitativa del estado del vehículo <V>
	Localización y retraso del vehículo <V> junto con las alarmas enviadas por el mismo
Qué podría pasar	
¿Qué podría pasar en la línea <L>?	Respuesta simple orientada a alertar al operador sobre la posible ocurrencia de una salida desatendida en alguna cabecera de línea
	Respuesta detallada con la predicción de la evolución de los vehículos retrasados y sus tiempos de llegada a las cabeceras de la línea
¿Qué podría pasar en la línea <L> si el vehículo <V> es enviado al garaje?	Respuesta simple orientada a informar al operador de la necesidad, o no, de llevar a cabo acciones de control como consecuencia de la disminución del número de vehículos en servicio
	Respuesta detallada con una estimación del efecto de la acción de control en la localización y retraso potencial del resto de vehículos en servicio
¿Qué podría pasar en la línea <L> si las condiciones del tráfico imponen una velocidad de <X> Km/h en el sector <S>?	Respuesta simple orientada a alertar al operador sobre la posible ocurrencia de una salida desatendida en algunas de las cabeceras de línea
	Respuesta detallada con la predicción de la evolución de los vehículos retrasados y sus tiempos de llegada a las cabeceras de línea, bajo el supuesto de una velocidad media de <X> Km/h en el sector <S>
¿Qué podría pasar con el horario del vehículo <V> de la línea <L> si las condiciones del tráfico imponen una velocidad de <X> Km/h en el sector <S>?	Respuesta simple orientada a alertar al operador sobre la posible ocurrencia de una salida desatendida del vehículo <V> desde su próxima cabecera de línea provocada por las condiciones del tráfico formuladas en la pregunta
	Respuesta detallada con la predicción de la evolución del vehículo <V>, su posible retraso y su tiempo de llegada al final de la línea, bajo el supuesto de una velocidad media de <X> Km/h en el sector <S>
Qué debería hacerse	
¿Qué debería hacerse en la línea <L>?	Conjunto de descripciones a alto nivel de acciones de control estratégicas capaces de mejorar una situación problemática observada
	Plan detallado de acciones de control atómicas que pueden resolver problemas observados
	Plan de control detallado junto con una valoración global de su impacto y eventos adicionales a considerar durante la vigilancia
¿Qué debería hacerse en la línea <L> si no hay vehículos de reserva en las áreas <A1, A2, ..., Am>?	Conjunto de descripciones a alto nivel de acciones de control estratégicas capaces de mejorar la situación problemática observada bajo las restricciones que impone el estado de las áreas de reserva
	Plan detallado de acciones de control atómicas que pueden resolver problemas observados en <L>



	Confirmación o rechazo de una acción de control sugerida con anterioridad.
	Plan de control detallado junto con una valoración global de su impacto y eventos adicionales a considerar durante la vigilancia
<i>Qué debería hacerse en la línea &lt;L&gt; si las condiciones del tráfico imponen una velocidad de &lt;X&gt; Km/h en el sector &lt;S&gt;?</i>	Conjunto de descripciones a alto nivel de acciones de control estratégicas capaces de manejar la situación provocada por las condiciones de tráfico planteadas en la pregunta
	Plan detallado de acciones de control atómicas que pueden resolver los problemas derivados de la situación impuesta por las condiciones del tráfico
	Confirmación o rechazo de una acción de control sugerida con anterioridad
	Plan de control detallado junto con una valoración global de su impacto y eventos adicionales a considerar durante la vigilancia
<i>Qué podría hacerse en la línea &lt;L&gt; con el vehículo &lt;V&gt; que está próximo a finalizar el servicio?</i>	Descripción a alto nivel de la posibilidad de emplear al vehículo <V> en alguna acción de control
	Descripción detallada de la forma de utilizar al vehículo <V> en alguna acción de control

Figura 7.5: Tipos de preguntas y respuestas para gestión de transporte público

7.3.1 Elementos del interfaz de usuario

La colección de preguntas que constituyen el modelo de interacción son traducidas al lenguaje de resolución de problemas por el interfaz de usuario. La forma en que este interfaz recibe dichas preguntas no es a través de un módulo especializado en la interpretación de expresiones en lenguaje natural, dado que éste no es el objetivo fundamental del sistema que se presenta, sino que se lleva a cabo mediante un sistema de menús.

Mediante este sistema, el usuario tiene acceso a un menú principal en el que aparecen las diferentes preguntas que éste puede hacer de acuerdo con el estado del diálogo, es decir, ese conjunto de preguntas varía dependiendo de las preguntas que se hayan hecho con anterioridad. La formulación de estas preguntas no es completa, en el sentido de que incluyen variables sobre parámetros que forman parte de la especificación de la pregunta que están asociadas a otros menús. Estos menús se despliegan al pinchar con el ratón sobre las variables y muestran los valores alternativos que dichas variables pueden tomar, y que pueden ser seleccionados también por el usuario con el teclado o el ratón.

Este sistema de menús se ilustra en las figuras 7.6, 7.7, 7.8, 7.9 y 7.10.



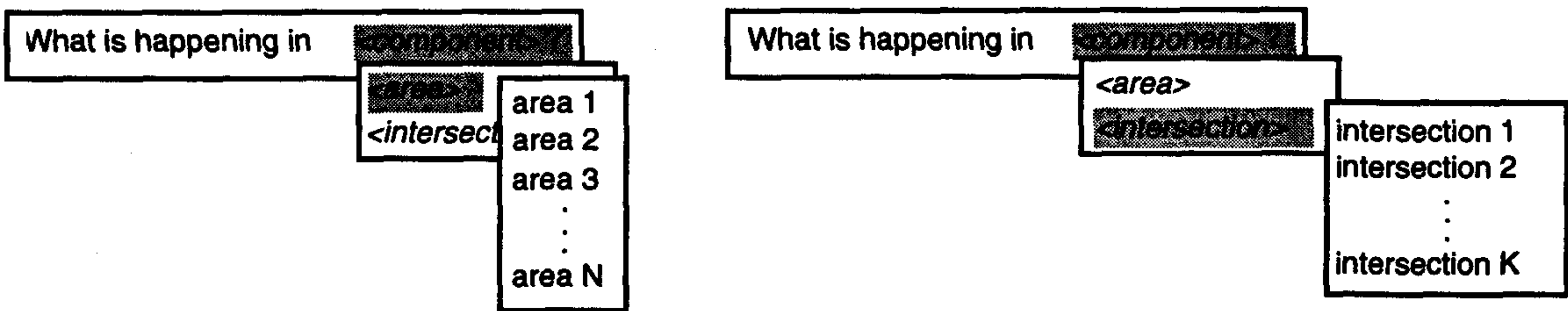


Figura 7.6: Formulación de preguntas del tipo *Qué está pasando*

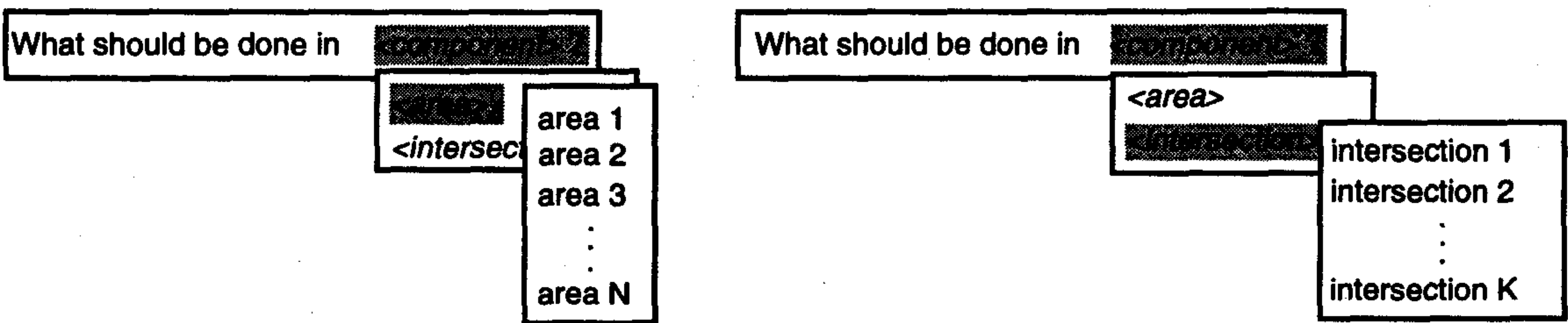


Figura 7.7: Formulación de preguntas del tipo *Qué debería hacerse*

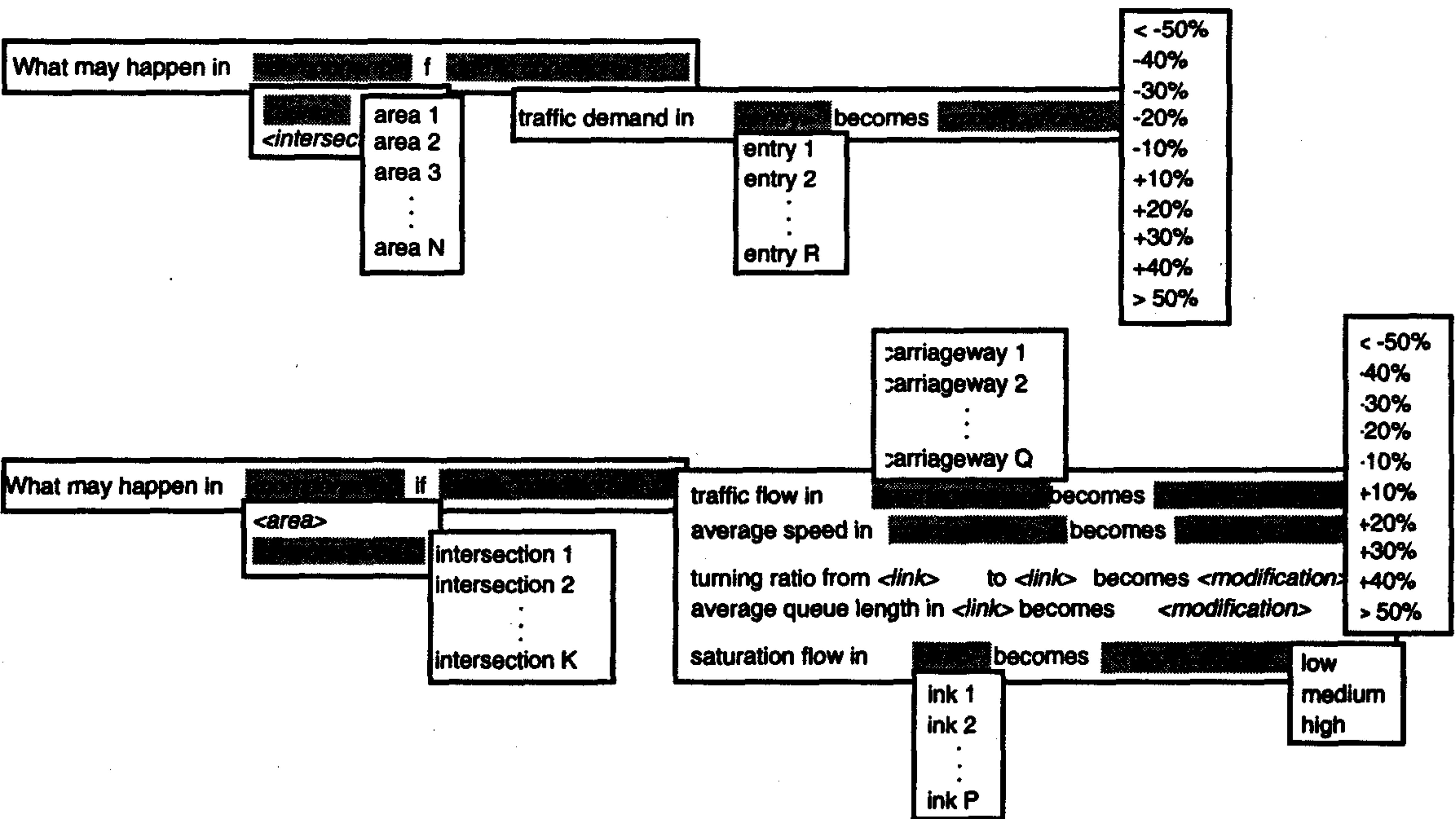


Figura 7.8: Formulación de preguntas del tipo *Qué pasaría si con hipótesis sobre condiciones del tráfico*

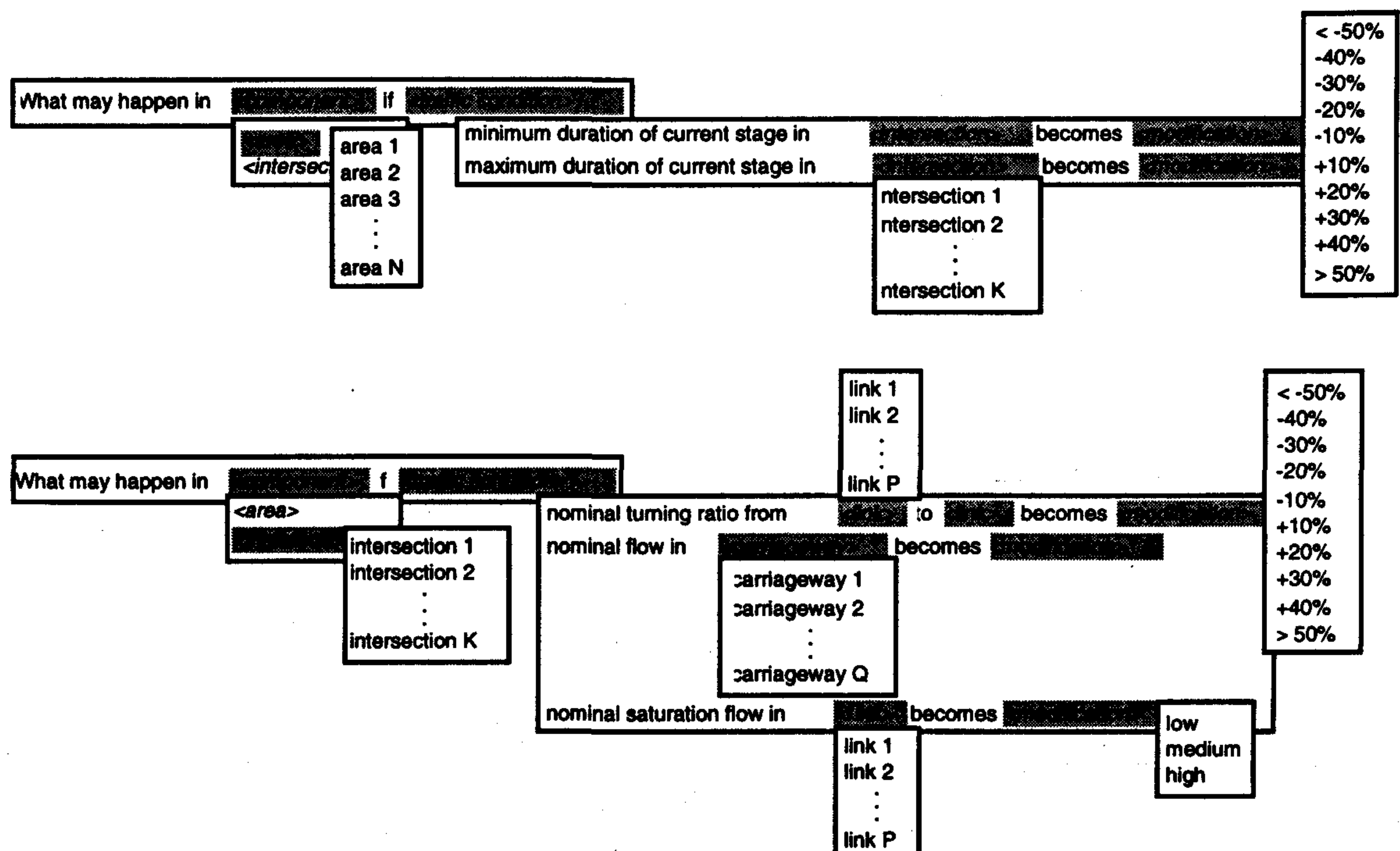


Figura 7.9: Formulación de preguntas del tipo *¿Qué pasaría si* con hipótesis sobre condiciones de control

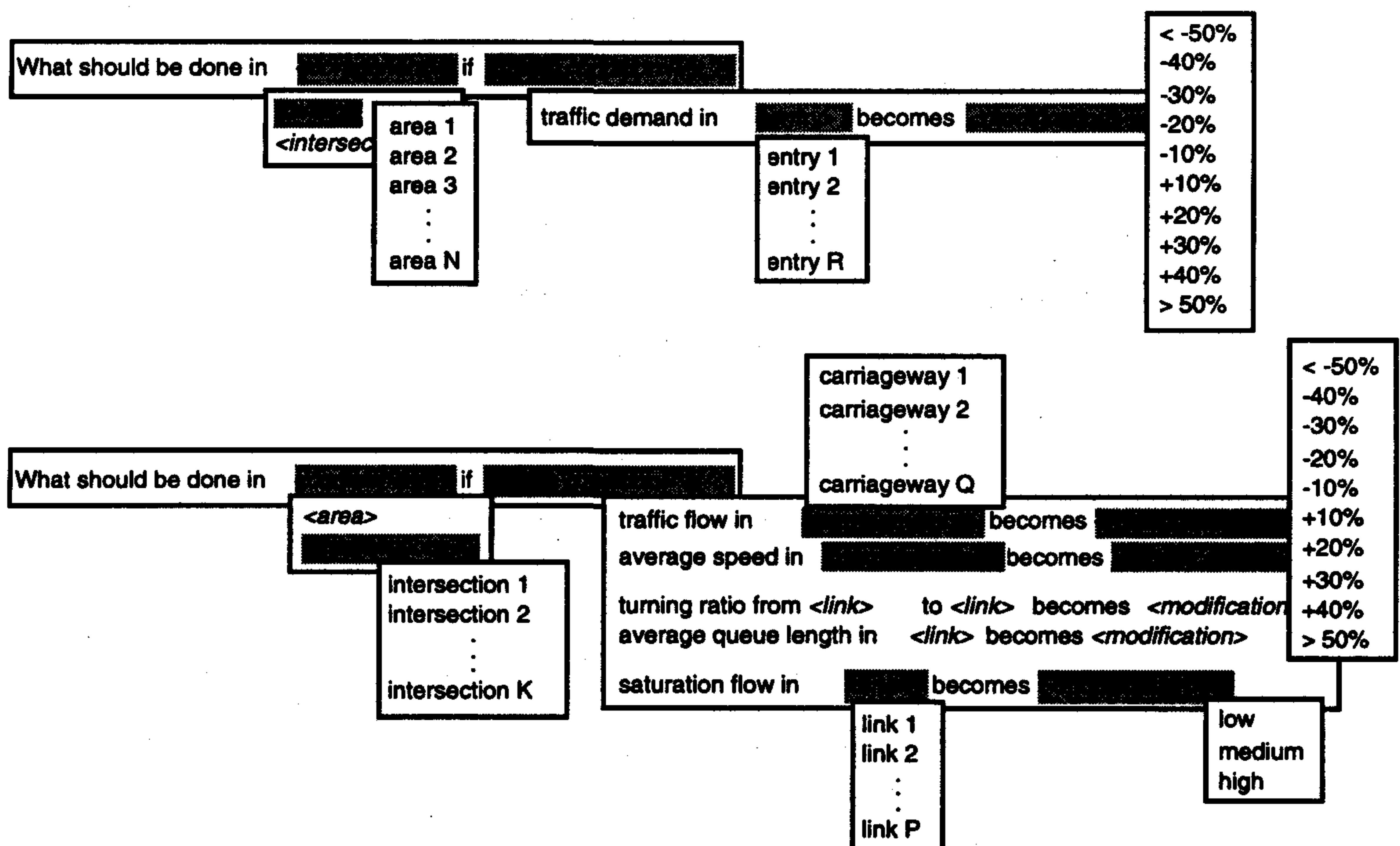


Figura 7.10: Formulación de preguntas del tipo *¿Qué debería hacerse* con hipótesis sobre condiciones del tráfico



## 7.4 El Espacio de Resolvedores de Problemas

De acuerdo con la descripción del problema de gestión de transporte público presentada en las secciones anteriores, el conocimiento de resolución de este tipo de problemas puede organizarse en tres grandes clases:

- *Conocimiento clasificativo.* Necesario para identificar la situación de una línea a partir de un conjunto de datos de estado básicos enviados por el sistema de información.
- *Conocimiento predictivo.* Empleado para estimar la evolución a corto plazo de los vehículos, y a partir de ellos de las líneas, para evaluar la gravedad de los problemas observados y proporcionar la información necesaria para corregir los retrasos de los vehículos involucrados en dichos problemas.
- *Conocimiento de planificación.* Usado para diseñar las acciones de control necesarias para resolver los problemas actuales o minimizar al menos sus consecuencias.

La visión funcional del modelo de conocimiento capaz de hacer uso de los tres tipos de conocimiento se presenta en las jerarquías de tareas y métodos descritas en las secciones siguientes. Dichas jerarquías se representan con árboles en cuyos niveles pares se definen tareas y en los impares métodos, hasta el nivel de las hojas que es par y está ocupado por el conocimiento declarativo que da soporte a los métodos básicos del nivel inmediatamente superior.

De acuerdo con la terminología empleada en las estructuras de tipo árbol, los nodos ocupados por tareas son nodos OR puesto que los métodos asociados con sus nodos descendientes definen modos alternativos de resolver la tarea, mientras que los nodos ocupados por métodos son nodos AND ya que las subtareas asociadas a sus nodos descendientes definen las actividades necesarias para llevar a cabo el método. En estos árboles, las tareas aparecerán incluidas en elipses y los métodos en rectángulos.

### 7.4.1 Resolución de problemas clasificativos

La identificación del estado de una línea ha de realizarse en dos pasos (ver figura 7.11):

- (1) Dado que la clasificación del estado de una línea depende completamente de los datos enviados por el sistema de información, un primer paso consiste en asegurarse de que la calidad de esos datos sea la adecuada. Esta tarea puede llevarse a cabo con dos métodos:
  - Uno que se limite a identificar errores o inconsistencias en el conjunto de datos.
  - Otro que además de identificar esos errores genere, si es necesario, valores razonables que suplan las carencias o inconsistencias detectadas.
- (2) Una vez validado el conjunto de datos de partida, se procede a clasificar el estado de la línea pudiendo emplear para ello tres tipos de métodos:
  - Uno basado en conocimiento heurístico que utiliza modelos declarativos de escenarios problemáticos en líneas de transporte público y que puede ofrecer soluciones a dos niveles de abstracción:
    - (i) Como un valor de una escala de alarmas usado para identificar estados globales de línea.
    - (ii) Como una descripción detallada de los elementos que determinan el estado actual de la línea, como la distribución y valor medio del retraso entre los vehículos de una línea, la presencia de vehículos averiados u obstáculos en la línea, etc.
  - Otro que amplía las funcionalidades del anterior ofreciendo información adicional a la estrictamente necesaria para caracterizar el estado de una línea, avisando sobre eventos previstos a corto plazo que podrían incidir en la gestión del



servicio, como un aumento de demanda o la finalización de la jornada por parte de algunos vehículos.

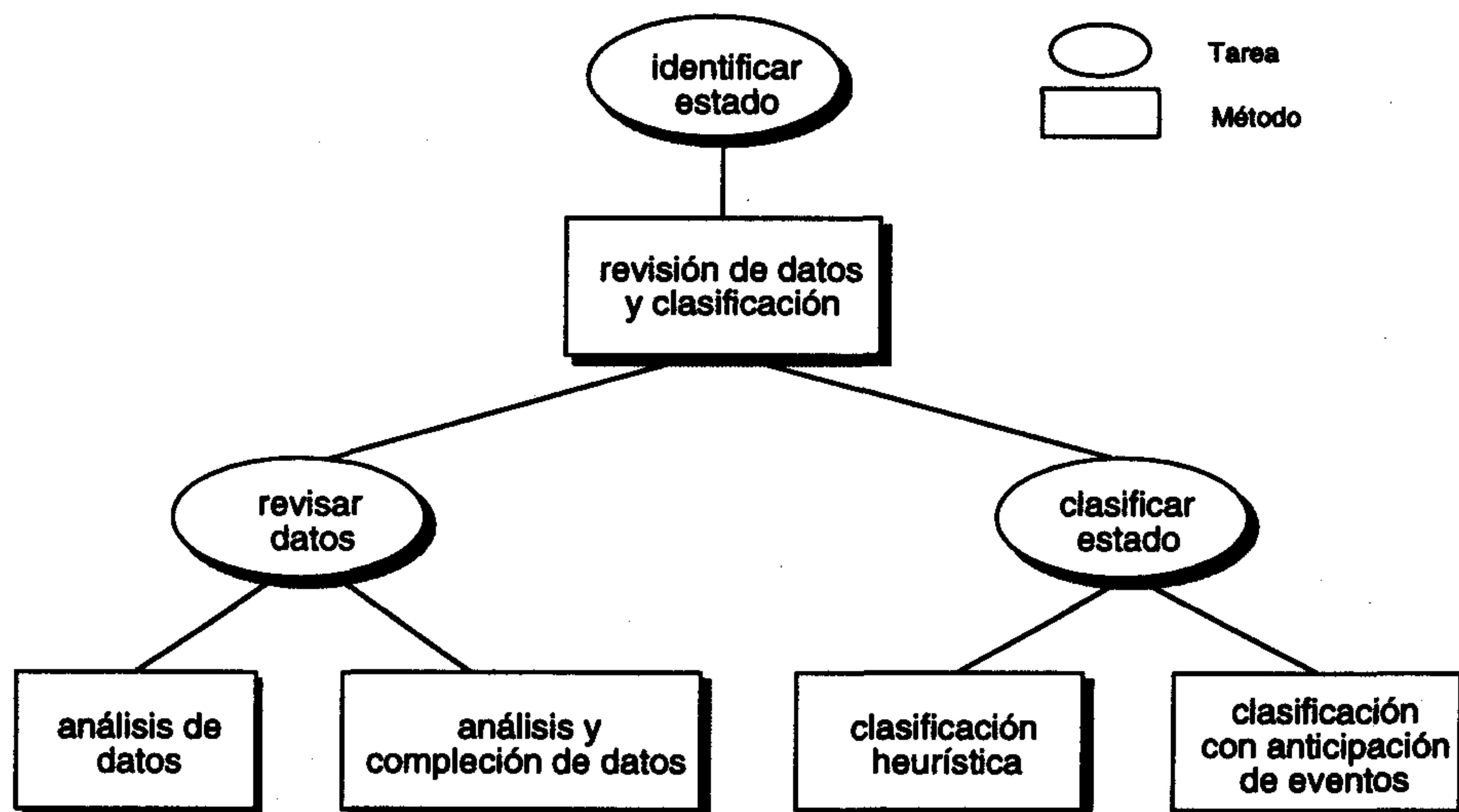


Figura 7.11: Estructura de tareas-métodos para la identificación de estados

#### 7.4.1.1 Revisión de datos

La tarea de revisar los datos enviados por el sistema de información puede hacerse, como se ha mencionado en el apartado anterior, con dos objetivos: (i) o bien con la única finalidad de detectar los datos erróneos y filtrar los válidos, lo que implica que el razonamiento para clasificar el estado de una línea deberá hacerse con un método capaz de manejar información incompleta, o bien (ii) generando además valores coherentes para aquellas variables para las que no se recibieron datos o éstos fueron erróneos.

La detección de errores consistirá simplemente en examinar las colecciones de datos para separar aquellos marcados con la etiqueta de erróneos por el sistema de información. En el caso en el que se lleve a cabo la tarea de completar los datos que faltan o son erróneos, ésta puede realizarse de tres formas (ver figura 7.12):

- Si el número de datos omitidos o erróneos es reducido, cabe la posibilidad de solicitar al usuario valores adecuados para los mismos.

- Aplicando una base de datos históricos de la cual pueden extraerse valores razonables observados bajo condiciones similares a las presentes cuando se plantea la pregunta. En este caso, el método realizaría una búsqueda en una colección de tablas que asocian tipos de días y franjas horarias con valores medios de los datos básicos de estado.
- Aplicando un método de compleción basado en un modelo de la línea, con el que se pueda simular la evolución de la línea a partir de un estado previo y extraer del mismo los datos requeridos.

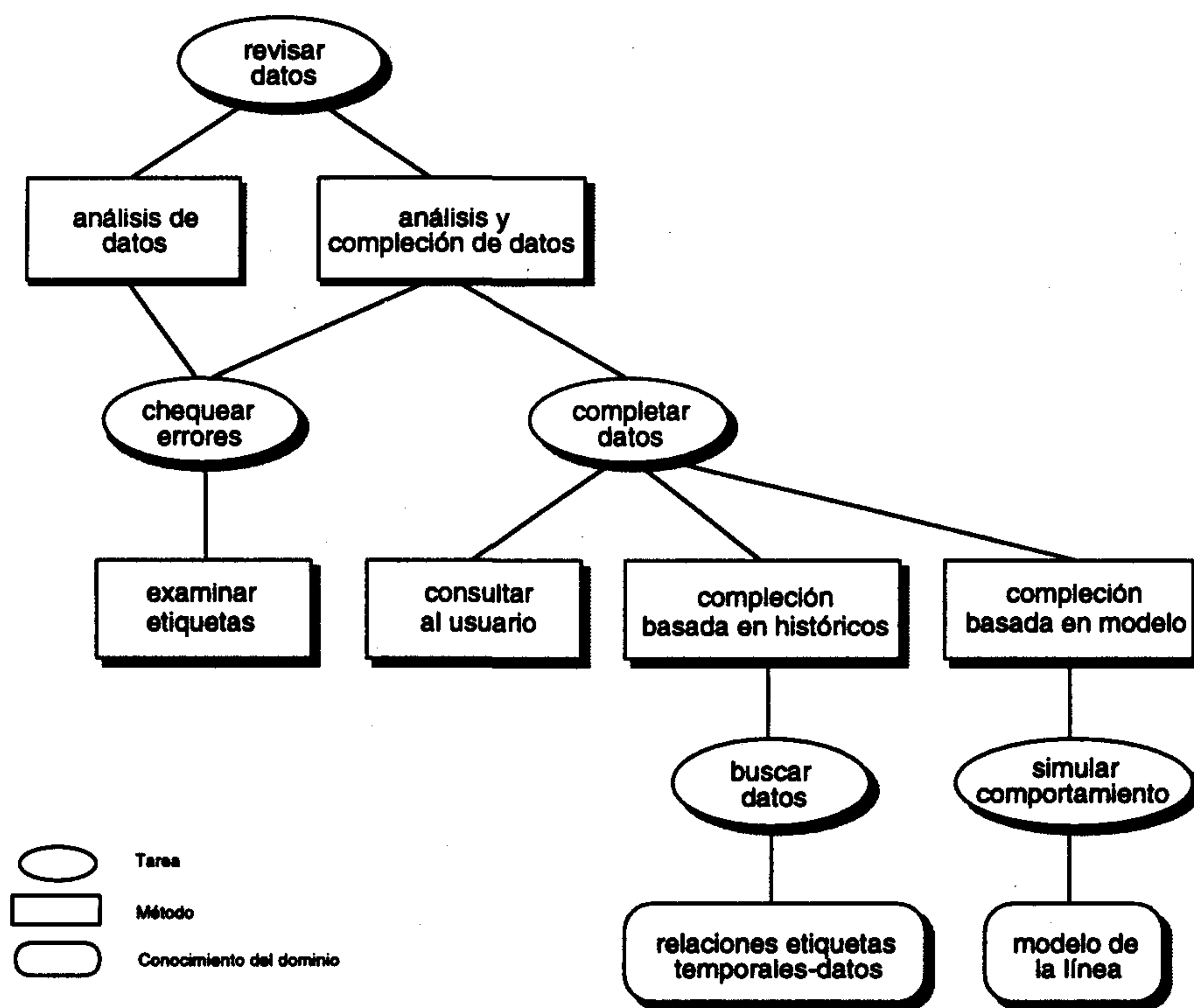


Figura 7.12: Estructura de tareas-métodos para la revisión de datos

#### 7.4.1.2 Clasificación de estados

Tal y como se mencionó anteriormente, la clasificación del estado de un componente del sistema, es decir, una línea o un vehículo, puede hacerse de dos maneras (ver figura 7.13):



- Atendiendo meramente a sus características actuales, aplicando un método de *clasificación heurística*, o bien
- Proyectándose de alguna manera hacia el futuro próximo avisando de eventos que pueden tener incidencia en el servicio, aplicando un método que podría denominarse *clasificación con anticipación*.

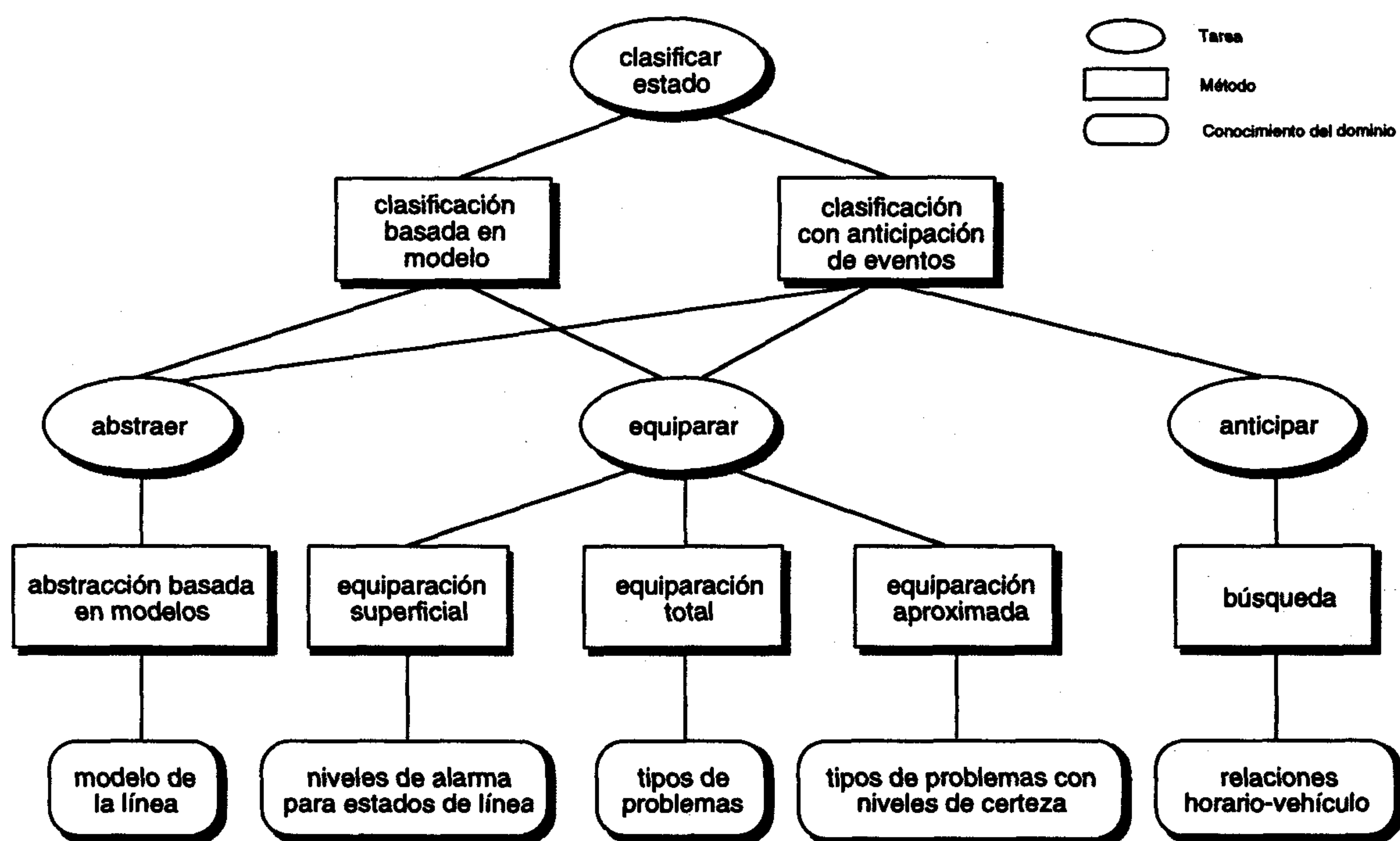


Figura 7.13: Estructura de tareas-métodos para la clasificación de estados

En ambos casos, es necesario disponer de datos de estado más elaborados que los que proporciona el sistema de información, por lo que ha de resolverse previamente una tarea de abstracción aplicando funciones estadísticas y modelos cualitativos para derivar valores abstractos, como la distribución y la media del retraso o una medida cualitativa de la dimensión del retraso, a partir de los datos básicos, como la posición de cada vehículo o su retraso en términos cuantitativos.

Los resultados de esta tarea de abstracción se emplean para llevar a cabo la tarea de equiparar estos datos con características de escenarios prototípicos, que puede realizarse a distintos niveles de detalle:

- Haciendo una clasificación superficial que se limita a ofrecer una valoración cualitativa muy general del estado de una línea.
- Aplicando un método de equiparación total con el que la identificación positiva de uno de estos escenarios como el que describe la situación observada en la línea se produce sólo si se da una coincidencia completa entre los datos observados y los que caracterizan el escenario.
- Aplicando un método de equiparación parcial, similar al anterior, pero que utiliza criterios con los que valorar el grado de similitud entre la situación descrita por los datos y la que describen los distintos escenarios.

Estos tres métodos utilizan un intérprete de reglas que aplica como procedimiento de inferencia el encadenamiento hacia delante. Además, en el caso del tercero, estas reglas tienen asociados factores de certeza que se utilizan aplicando técnicas tipo MYCIN.

La figura 7.14 muestra la estructura completa de tareas y métodos con los que resolver problemas clasificativos.

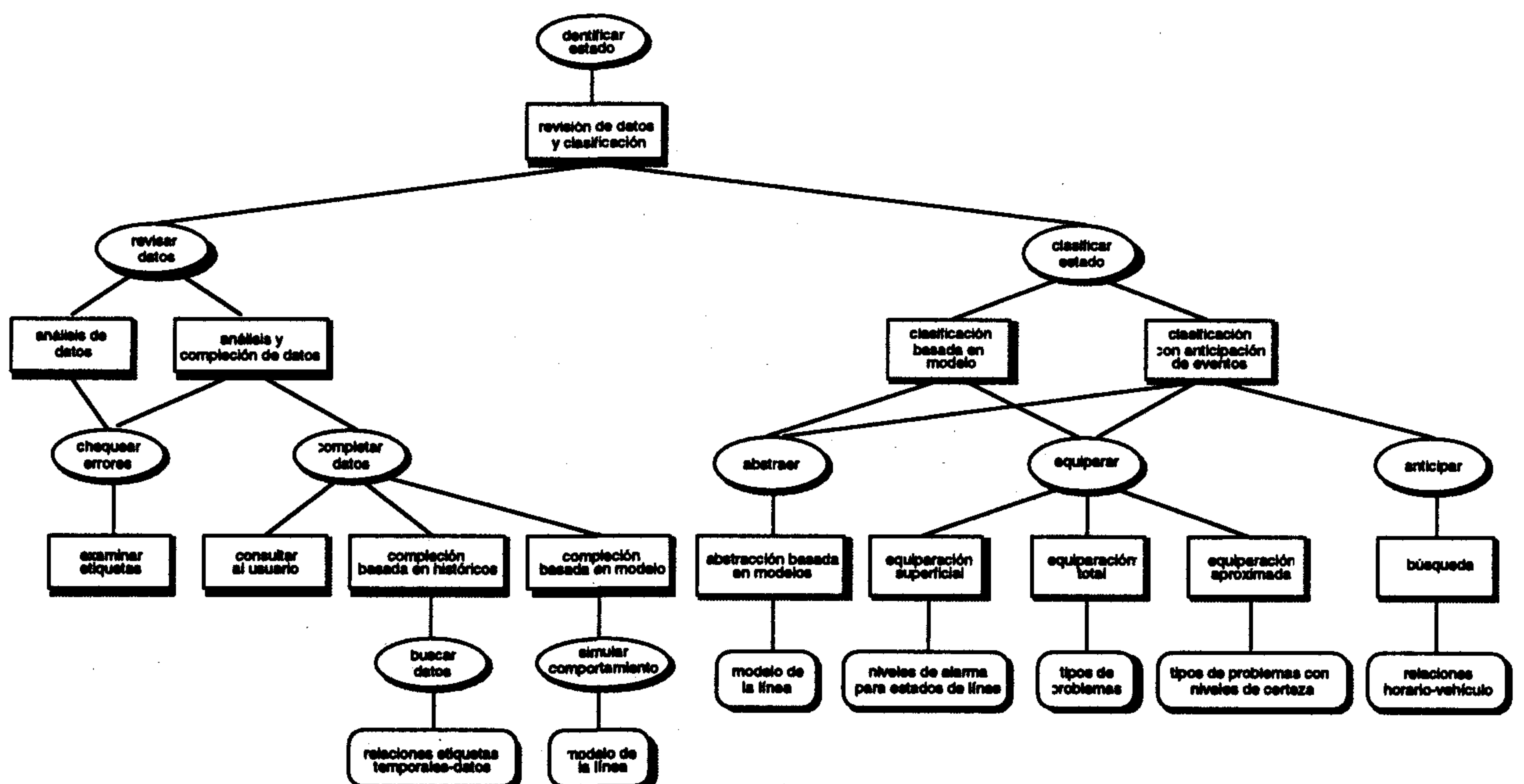


Figura 7.14: Estructura tareas-métodos para la solución de problemas clasificativos



## 7.4.2 Resolución de problemas predictivos

De forma similar a la descripción del estado actual de una línea, la estimación del estado que en un futuro próximo tendrá un componente del sistema puede ofrecerse a dos niveles de abstracción:

- Dando una valoración global de la calidad del servicio a corto plazo obtenida de un modelo heurístico de predicción.
- Con una descripción detallada del comportamiento esperado y posición de los vehículos retrasados obtenida de un modelo de comportamiento de vehículos.

Para dar esta segunda clase de respuesta se necesitará utilizar un modelo de la línea y la evolución observada de los vehículos para analizar si el servicio planificado difiere o no del que previsiblemente ofrezcan estos vehículos a corto plazo, identificando así el estado a corto plazo de la línea.

Entonces, la predicción del estado de una línea puede llevarse a cabo con dos tipos de métodos (ver figura 7.15):

- Uno clasificativo que utiliza un modelo de predicción heurística, definido a partir de la experiencia de los operadores, que asocia la presencia de eventos relevantes observados con estados futuros previsibles; y
- Uno más complejo basado en modelos en el que se llevan a cabo cuatro subtareas:
  - (1) Estimación de los tiempos medios de recorrido de los vehículos en los distintos sectores de la línea utilizando información histórica sobre el comportamiento de las líneas o bien mediante la extrapolación de valores de estado de la línea obtenidos recientemente.
  - (2) Simulación de la evolución a corto plazo de los vehículos retrasados usando modelos abstractos de las líneas que incluyen su descripción topológica.

- (3) Extracción de una base de datos del horario previsto para los vehículos retrasados.
- (4) Comparación de los estados esperado y previsible de los vehículos retrasados para inferir posibles cortes en el servicio y la gravedad de los mismos.

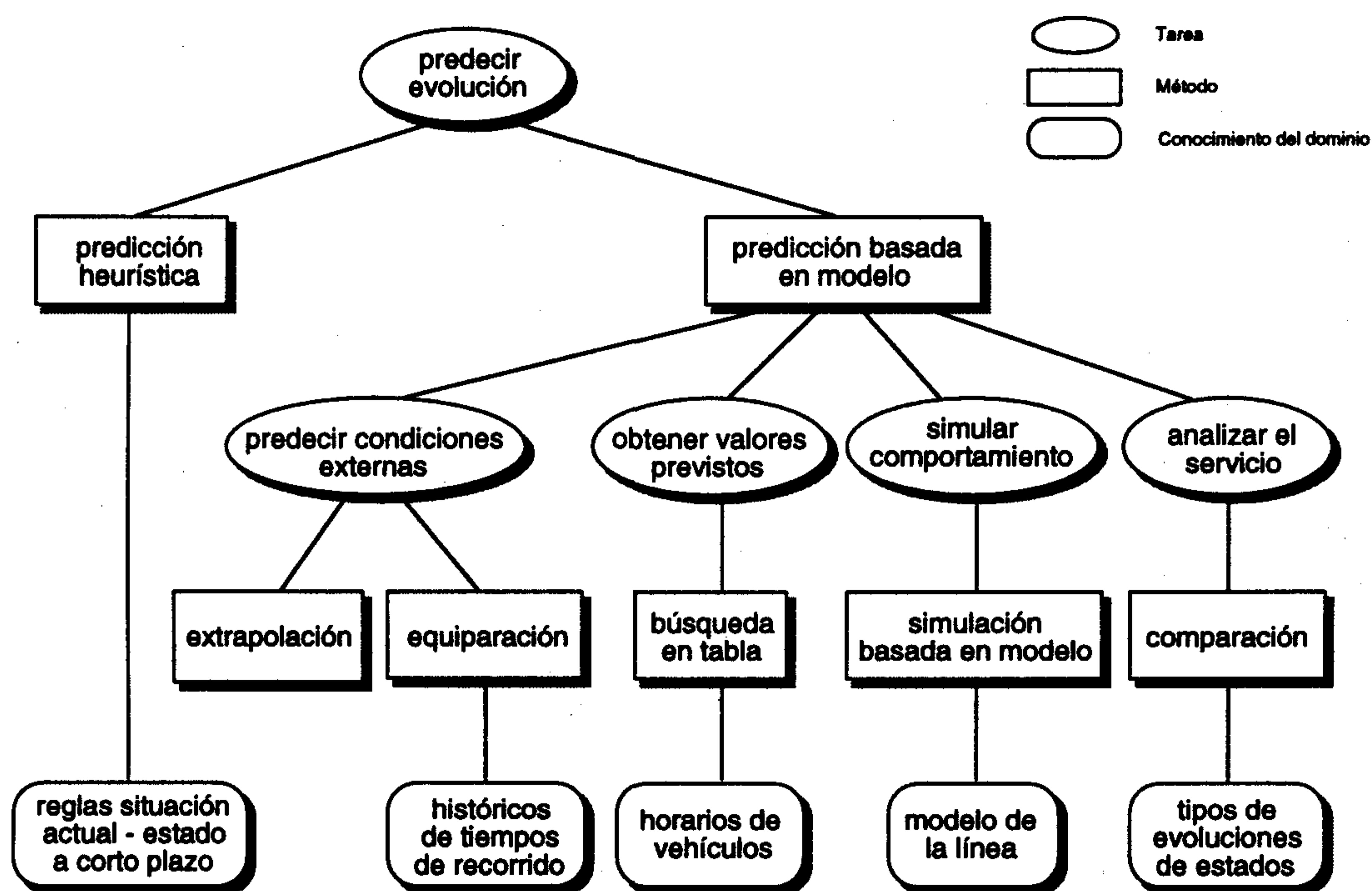


Figura 7.15: Estructura de tareas-métodos para la estimación de estados futuros

Además de los métodos anteriores, es necesario añadir un método con el que se redefinan las condiciones de estado de una línea, utilizadas como entrada para hacer la predicción, cuando se plantean preguntas que introducen hipótesis sobre algunas de esas condiciones y modifican su valor actual conocido, es decir, para preguntas del tipo *qué pasaría si*. Dicho método llevaría a cabo entonces dos subtareas:

- (i) Primero modificaría los parámetros descriptivos del estado actual de la línea en función de las condiciones de entorno declaradas.
- (ii) Después llevaría a cabo la evaluación del estado previsto a corto plazo llamando al método de predicción descrito anteriormente.



### 7.4.3 Resolución de problemas de planificación

El conocimiento para generar propuestas de acciones de control en una línea puede entenderse dividido en dos clases:

- Un conocimiento superficial que establece relaciones directas entre tipos de problemas y soluciones, que resulta aplicable sobre problemas sencillos y bien conocidos para los que la acción a tomar está predeterminada y de la que sólo es necesario especificar algunos parámetros. Por ejemplo, para retrasos inferiores a 5 minutos aplica una acción de microcontrol que reduce automáticamente el tiempo de descanso de los autobuses en la cabecera de línea en la medida necesaria para recuperar el servicio.
- Un conocimiento de planificación más detallado que está distribuido jerárquicamente entre un conjunto de especialistas en planificación de acciones de control definidos a diferentes niveles de abstracción. El especialista de más alto nivel es capaz de ofrecer soluciones expresadas en términos estratégicos que pueden ser refinadas y detalladas con la ayuda de especialistas de más bajo nivel. La configuración de un plan de control se gestiona desde un módulo especializado que contiene el conocimiento necesario para decidir el tipo de especialista requerido para detallar cada uno de los pasos de un plan de control. Por su parte, cada especialista dispone de tres clases de conocimiento:
  - ◊ Conocimiento para decidir su aplicabilidad bajo las condiciones impuestas por el contexto del proceso de razonamiento.
  - ◊ Una vez establecida la aplicabilidad, conocimiento para seleccionar la clase de solución que debe aplicarse entre un conjunto de alternativas.
  - ◊ Para cada plan alternativo, conocimiento sobre la descomposición en acciones atómicas y/o subplanes que definen el plan.

Ahora, la tarea de generar propuestas de acciones de control que recuperen o mejoren el nivel de servicio de las líneas puede llevarse a cabo con los siguientes métodos:

- Usando un método que se limita a elaborar un plan de acción para la situación detectada actualmente en la línea.
- Con un método que además de ofrecer una solución para el estado actual avisa sobre eventos a corto plazo que podrían influir en la ejecución del plan, de forma similar al método descrito en el apartado 7.4.1.2 que avisaba de eventos a corto plazo que podrían influir en el estado del línea.
- Usando un método que primero predice la situación de la línea a corto plazo - solicitando la resolución de la tarea *predecir estado*, y luego utiliza esta información para elaborar el plan de acciones a tomar.
- Con un método completo que predice el estado de la línea a corto plazo y elabora un plan de actuación, al igual que el anterior, y además notifica eventos a corto plazo con influencia en la evolución del servicio.

Los cuatro métodos anteriores necesitan resolver la tarea de elaborar un plan de control, para la que pueden emplearse dos posibles métodos:

- Un método superficial, que utiliza el conocimiento superficial de control mencionado más arriba y que está soportado por una base de conocimiento de reglas que relacionan posibles escenarios problemáticos de línea con planes de control; y
- Un método detallado que emplea la estrategia de *establecer & refinar* con el conjunto de especialistas en la configuración de diferentes aspectos de un plan de control. Este método hace una búsqueda jerárquica basada en un bucle de tres etapas, llevado a cabo por subtareas más básicas, hasta la especificación final del plan de control. Las subtareas asociadas a este método son tres:

- (1) Primero, el especialista analiza si la entrada satisface las condiciones de aplicabilidad del especialista. Inicialmente, se selecciona el especialista de más alto nivel en la jerarquía y sus condiciones de aplicabilidad se consideran satisfechas por defecto.



- (2) Entonces, el conocimiento de refino del especialista se aplica para identificar cuál de las posibles soluciones/acciones de control de que dispone es la más adecuada para ser adoptada.
- (3) Finalmente, se analizan las distintas etapas de la solución elegida anteriormente para identificar aquellas cuya realización requiere la intervención de otros especialistas.



*Figura 7.16: Estructura de tareas-métodos para la generación de acciones de control sin predicción previa del estado futuro*

Las otras subtareas de los cuatro métodos mencionados serían realizadas por los correspondientes métodos de predicción y anticipación de eventos mencionados en las secciones anteriores.

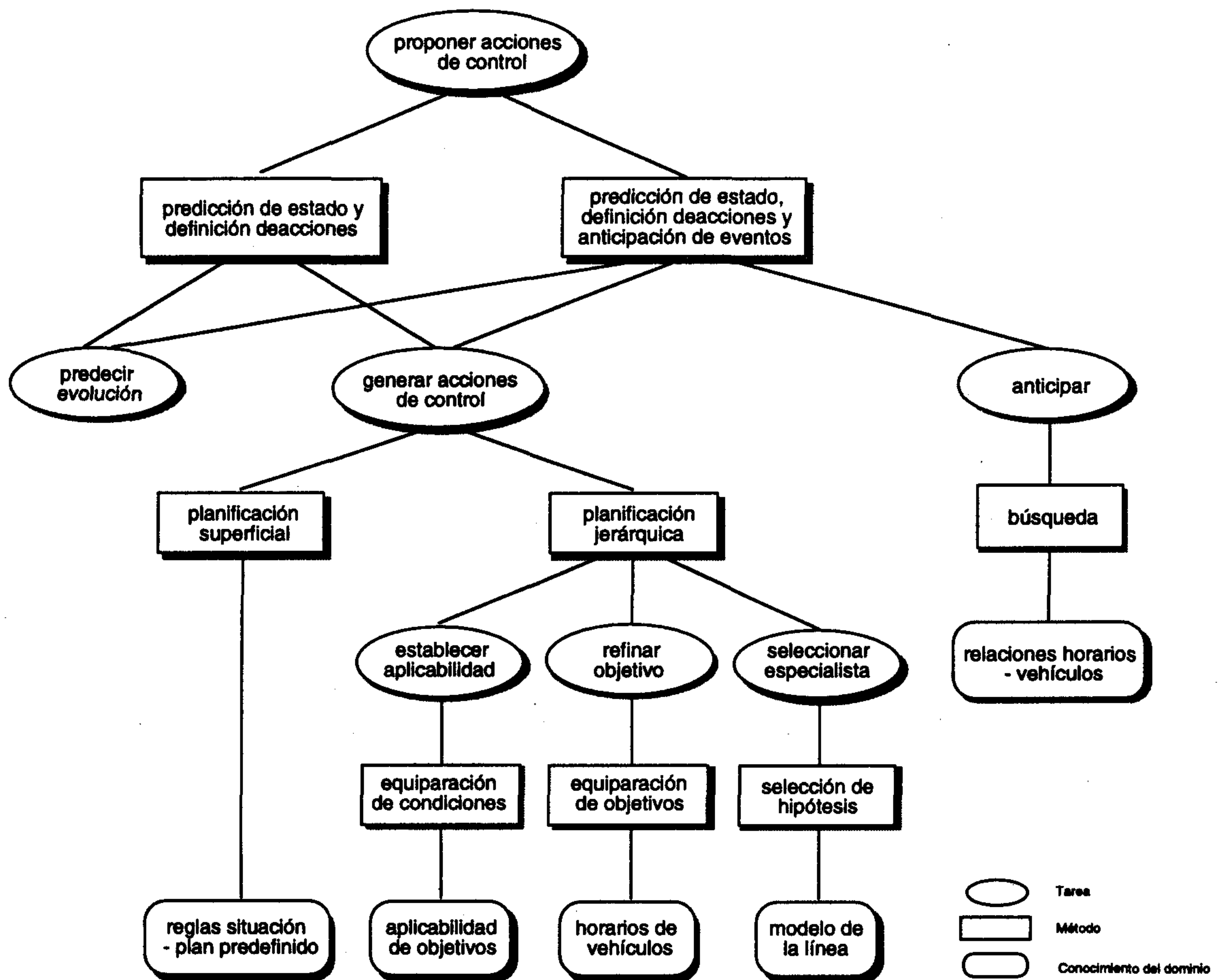


Figura 7.17: Estructura de tareas-métodos para la generación de acciones de control con predicción previa del estado futuro

## 7.5 El Razonador Metanivel

### 7.5.1 Modelo de escenarios de interacción

La especificación del modelo de diálogo entre el sistema de gestión de transporte público y sus usuarios requiere:

- Identificar los distintos tipos de usuarios que puede tener el sistema.
- Caracterizar para cada uno de ellos el modo en el que el sistema se comunicará con ellos, es decir, qué tipo de preguntas puede hacer cada uno de ellos y qué clase de respuestas pueden obtener.



Tras el análisis del dominio problema, la clase de usuarios del sistema y los modelos de interacción con cada uno de ellos son los siguientes:

### Supervisor

Es responsable de lograr un funcionamiento global satisfactorio del sistema de transporte público. Esto implica:

- (i) Asegurarse de que los problemas detectados en las líneas son adecuadamente atendidos por los operadores.
- (ii) Distribuir los recursos de control disponibles (los vehículos de reserva y reparaciones) en la forma más provechosa para el conjunto de las líneas.
- (iii) Supervisar que no se produzcan interacciones indeseables en áreas de la red compartidas por varias líneas debido a las acciones de control sugeridas por los distintos operadores de esas líneas.

Entonces, este tipo de usuario busca fundamentalmente adquirir una descripción del estado general de la red o de un subconjunto de sus líneas, en la que se destaquen los aspectos más relevantes desde un punto de vista estratégico. Asimismo, le es de utilidad el acceso a la clase de recomendaciones propuestas por el sistema a los operadores de las diferentes líneas con problemas para evaluar la conveniencia de aplicarlas simultáneamente, evitando las interacciones no deseadas, y distribuyendo los recursos de control disponibles de la forma más adecuada. Por tanto, aunque normalmente el tipo de información requerida por este usuario será de tipo general, puesto que ha de supervisar la gestión de un gran número de líneas simultáneamente, también ha de tener la posibilidad de acceder a las respuestas detalladas ofrecidas a los operadores, tanto del estado de líneas concretas como de los pasos necesarios para llevar a cabo las acciones recomendadas.

Por tanto, se entiende que un usuario de la clase supervisor puede comunicarse con el sistema con dos objetivos:

- Para vigilar de forma general el estado de una o varias líneas. En este caso, le bastaría con dos de las tres grandes clases de preguntas - qué está pasando y qué debería hacerse, sin emplear su formulación

condicional. Con este objetivo puede solicitar al sistema que active el modo de funcionamiento en alerta sobre las líneas que no requieren su atención, para poder dedicarse a gestionar en detalle líneas con mayores problemas.

- Para realizar una monitorización detallada de una o varias líneas. En este caso, podría llegar a utilizar cualquiera de los tipos de preguntas que admite el sistema dado que además dispone del nivel máximo de privilegios de acceso al sistema.

Con respecto al modo en el que el sistema ofrecerá sus respuestas a este usuario, es decir, los requisitos de interacción que tomará como referencia para hacer la selección de los correspondientes métodos de resolución de problemas:

- Si el sistema entiende que el objetivo es la vigilancia entonces buscará métodos que generen respuestas a nivel abstracto, que para obtenerlas no requieran solicitar directamente al usuario información adicional, y que no necesiten consumir demasiados recursos. En términos de los atributos de interacción, éstos tomarían por defecto los siguientes valores: nivel de abstracción = alto, consumo limitado de recursos = si, madurez de la información = histórica, nivel de precisión = bajo, consulta al usuario = no.
- Si el sistema entiende que el objetivo es la monitorización entonces buscará métodos que den respuestas a un nivel de abstracción inferior al anterior pero en este caso sí se permitiría solicitar al usuario información adicional. Los atributos de interacción quedarían entonces: nivel de abstracción = medio, consumo limitado de recursos = si, madurez de la información = no, nivel de precisión = bajo, consulta al usuario = si.

### Operador

Es responsable de la gestión detallada y simultánea de varias líneas (entre 3 y 5 dependiendo de su complejidad), lo que implica:

- (i) Recopilar toda la información necesaria para caracterizar completamente la situación de cada una de las líneas que controla.



- (ii) Elaborar planes de actuación sobre líneas con problemas que ayuden a recuperar el nivel de servicio previsto.
- (iii) Comunicarse con los conductores de los vehículos tanto para obtener información complementaria de su estado como para notificarles las actuaciones que les corresponden dentro del plan de control.

Por tanto, los operadores necesitan disponer de información detallada sobre el estado de los vehículos y la presencia de eventos inesperados, así como toda la información que pueda proporcionarle el sistema orientada a la recuperación del nivel de servicio adecuado en las líneas con problemas. Sin embargo, al gestionar simultáneamente varias líneas, se considera necesario poder ofrecerles la información de estado a distintos niveles de especificidad, para que puedan centrar su atención en la gestión de una línea especialmente problemática sin descuidar a las demás, de las que pueden recibir una impresión general que les permita percatarse de los cambios de estado que podrían requerir su atención. De forma similar, el grado de detalle con el que expresar los planes de acciones de control sugeridos por el sistema no tiene porqué ser único, ya que puede depender de la experiencia del operador o de la atención que haya podido prestar al proceso de generación de dichos planes.

Entonces, se entiende que un usuario de la clase operador puede comunicarse con el sistema con dos objetivos:

- Para vigilar de forma general el estado de una o varias líneas, de manera análoga al supervisor, por lo que también le bastaría con la versión no condicional de las tres clases de preguntas - qué está pasando, qué podría pasar y que debería hacerse. Este usuario también puede solicitar al sistema que active el modo de funcionamiento en alerta sobre las líneas que no requieren su atención para poder dedicarse a gestionar en detalle líneas con mayores problemas.
- Para realizar una gestión detallada de una o varias líneas, para lo que podría llegar a utilizar cualquiera de los tipos de preguntas que admite el sistema salvo aquellas que se refieren a líneas que no están bajo su control.

Con respecto al modo en el que el sistema ofrecerá sus respuestas a este usuario, es decir, los requisitos de interacción que tomará como referencia para hacer la selección de los correspondientes métodos de resolución de problemas:

- Si el sistema entiende que el objetivo es la vigilancia entonces buscará métodos que generen respuestas a nivel abstracto, que para obtenerlas no requieran solicitar directamente al usuario información adicional, pero que se den basándose en la información más reciente y en el menor tiempo posible sin limitar a priori, por tanto, el consumo de recursos. En términos de los atributos de interacción, éstos tomarían por defecto los siguientes valores: nivel de abstracción = alto, consumo limitado de recursos = no, madurez de la información = reciente, nivel de precisión = alto, consulta al usuario = no.
- Si el sistema entiende que el objetivo es la monitorización entonces buscará métodos que den respuestas detalladas y precisas, con datos recientes y en poco tiempo, además de que sí se permitiría solicitar al usuario información adicional. Los atributos de interacción quedarían entonces: nivel de abstracción = bajo, consumo limitado de recursos = no, madurez de la información = reciente, nivel de precisión = alto, consulta al usuario = si.

En ambos casos, podría introducirse un matiz mediante el que se distinguirían los operadores expertos de los inexpertos, puesto que los primeros pueden prescindir en muchas ocasiones de las explicaciones que puede proporcionarles el sistema al conocer bien el funcionamiento de las líneas y el sistema, y poder entender por sí mismos cuáles son las razones que justifican sus respuestas.

### Observador

No tiene responsabilidades sobre la gestión de las líneas, estando su papel limitado al de un usuario circunstancial en busca de información. Por ejemplo, un observador podría ser un individuo que consulta al sistema para obtener la información que necesita para una emisión radiofónica en la que se ofrece a los oyentes una descripción del estado de la red de transporte público.

Por tanto, este tipo de usuario busca obtener meramente una impresión del funcionamiento de una o más líneas y no requiere información detallada ni



tiene acceso a las estrategias de control empleadas para resolver situaciones problemáticas. Eso implica que el sistema sólo le va a dar acceso a las preguntas sobre el estado actual o previsto a corto plazo para una o más líneas.

Los métodos empleados en la generación de las respuestas para este tipo de usuario se seleccionarían con criterios de mínimo coste, es decir, métodos que no consumen recursos y que pueden apoyarse en información histórica para elaborar respuestas ya que este tipo de usuario no conoce el dominio en profundidad por lo que no puede solicitársele información adicional. Además, las respuestas se formularían en términos abstractos y sin obsesionarse por la precisión. Expresado con los modos de interacción, este comportamiento se lograría con: nivel de abstracción = alto, consumo limitado de recursos = si, madurez de la información = histórica, nivel de precisión = bajo, consulta al usuario = no.

### 7.5.2 Modelo reflexivo de resolución de problemas

Como se presentó en el capítulo 6, el conocimiento reflexivo está distribuido entre una colección de especialistas. En el caso en el que uno de estos especialistas disponga de más de un método para realizar su tarea, la decisión sobre cuál de ellos es más apropiado para generar una respuesta en un momento dado de un diálogo usuario-sistema está guiada por los requisitos de interacción establecidos para el usuario y su pregunta, y el estado de la memoria de la interacción.

No todas las tareas descritas en el Espacio de Resolvedores de Problemas para gestionar una red de transporte público tiene más de un método asociado. Por esta razón, la descripción del modelo reflexivo se centrará en aquellos especialistas que sí disponen de más de un método para mostrar cómo hacen la selección. De acuerdo con el contenido del espacio de Resolvedores de Problemas, los especialistas de este tipo pueden clasificarse en tres grupos:

#### **(1) Especialistas de tareas asociadas a preguntas del tipo *¿qué está pasando?***

- ◆ Especialista en la tarea *revisar datos*



Como se vio en el apartado 7.4.1.1, la revisión de datos se puede hacer de dos formas:

- (i) Sólo detectando los datos erróneos para que no se utilicen en procesos posteriores de inferencia, o
- (ii) Detectando los errores y generando nuevos valores para esos datos incorrectos.

La decisión sobre cuál de ellos aplicar se toma como sigue:

- Si el nivel de asistencia que requiere el usuario por parte del sistema es bajo y la respuesta hay que generarla limitando el consumo de recursos entonces se elige el método de *chequeo de datos*,
- En otro caso se elige el método *de chequeo & completación*.

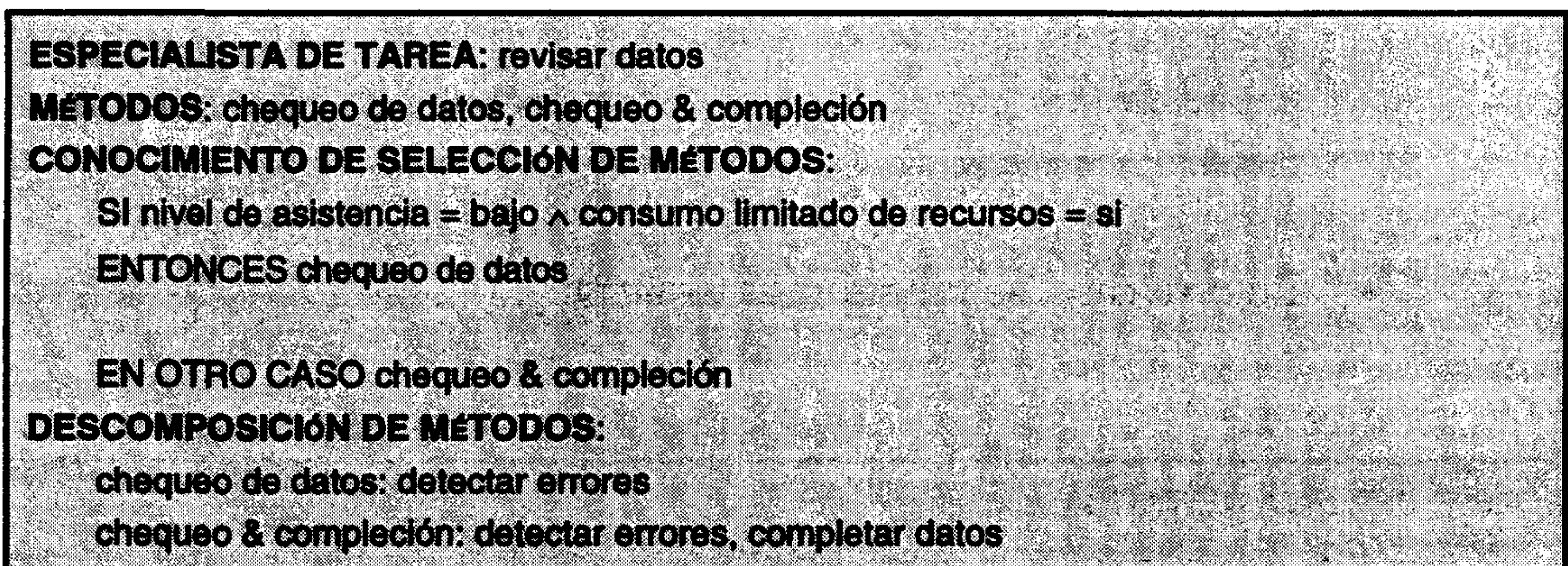


Figura 7.16: Especialista en la tarea de revisar\_datos

#### ◆ Especialista en la tarea *completar datos*

La tarea de completar datos se puede hacer también con cuatro métodos:

- (i) Solicitando al usuario un valor coherente para esos datos erróneos,
- (ii) Utilizando bases de datos históricas de las que aproximar los datos actuales,



- (iii) Aplicando métodos matemáticos de interpolación, o
- (iv) Aplicando un método de compleción basado en modelos.

Para decidir entre ellos:

- Si el nivel de asistencia es alto, o no es posible consultar al usuario, o el número de errores es significativo, y no se requiere la capacidad de generar explicaciones, y se pide utilizar la información disponible más reciente entonces se elige la *compleción por interpolación*.
- Si el nivel de asistencia es alto, o no es posible consultar al usuario o el número de errores es significativo, y no se requiere la capacidad de generar explicaciones, y no se pide utilizar la información disponible más reciente entonces se elige la *compleción con históricos*.
- Si el nivel de asistencia es alto, o no es posible consultar al usuario, o el número de errores es significativo, y sí se requiere la capacidad de generar explicaciones entonces se elige la *compleción basada en modelos*.
- En otro caso, se pregunta al usuario.

**ESPECIALISTA DE TAREA:** completar datos

**MÉTODOS:** consulta datos al usuario, compleción por interpolación, compleción con históricos, compleción basada en modelos

**CONOCIMIENTO DE SELECCIÓN DE MÉTODOS:**

SI (nivel de asistencia = alto  $\vee$  consulta al usuario = no  $\vee$  porcentaje de datos erróneos  $> 10\%$ )  $\wedge$  capacidad de explicación = no  $\wedge$  madurez de la información = reciente

ENTONCES compleción por interpolación

SI (nivel de asistencia = alto  $\vee$  consulta al usuario = no  $\vee$  porcentaje de datos erróneos  $> 10\%$ )  $\wedge$  capacidad de explicación = no  $\wedge$  madurez de la información = histórica

ENTONCES compleción con históricos



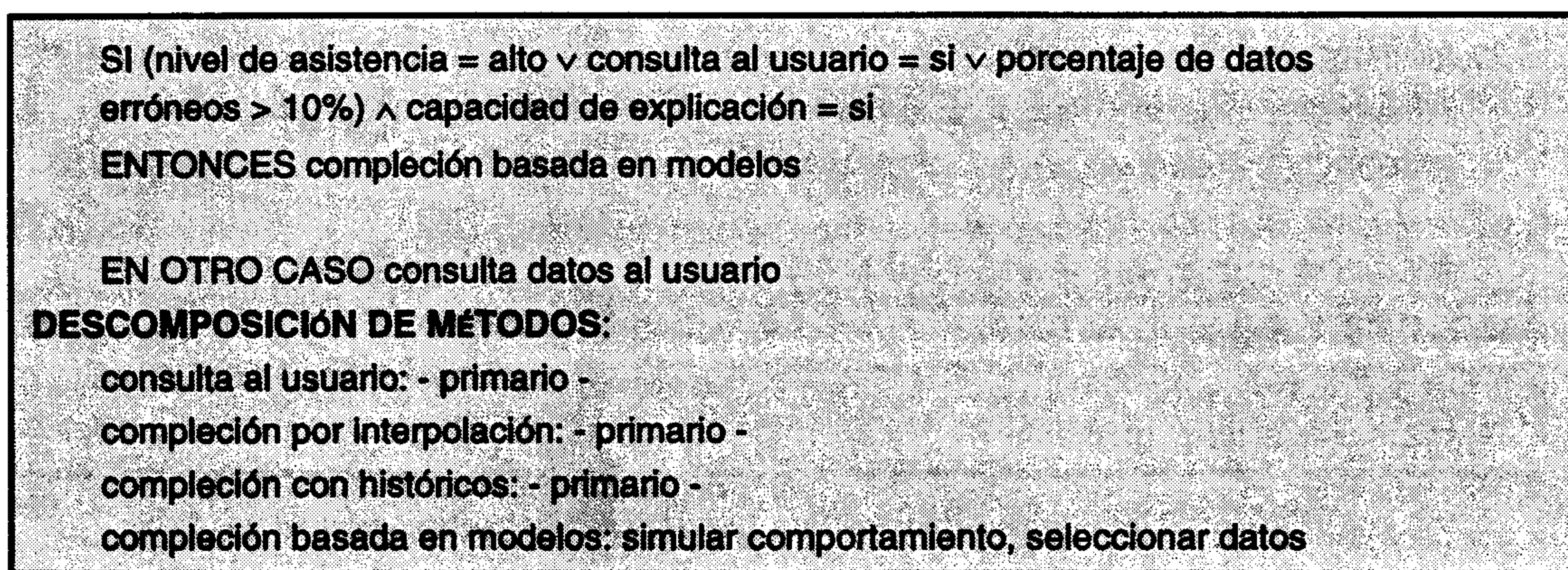


Figura 7.17: Especialista en la tarea de completar\_datos

◆ Especialista en la tarea *simular comportamiento*

Si el especialista en la tarea de completar datos elige el método de compleción basada en modelos, la realización de ese método requiere la intervención del especialista en la tarea de simular el comportamiento de una línea. Esta simulación se puede hacer de dos formas:

- (i) Con un modelo de macrosimulación que calcula parámetros del comportamiento de la línea en sus distintas secciones que pueden utilizarse para deducir de forma aproximada los datos que se necesitan completar.
- (ii) Con un modelo de microsimulación que reproduce la evolución de cada uno de los autobuses de una línea para obtener directamente los datos requeridos.

La decisión entre los dos:

- Si se tienen restricciones de tiempo y no se pretende obtener datos muy precisos entonces se elige el *macrosimulador*.
- En otro caso, se elige el *microsimulador*.

**ESPECIALISTA DE TAREA:** simular comportamiento  
**MÉTODOS:** macrosimulación, microsimulación



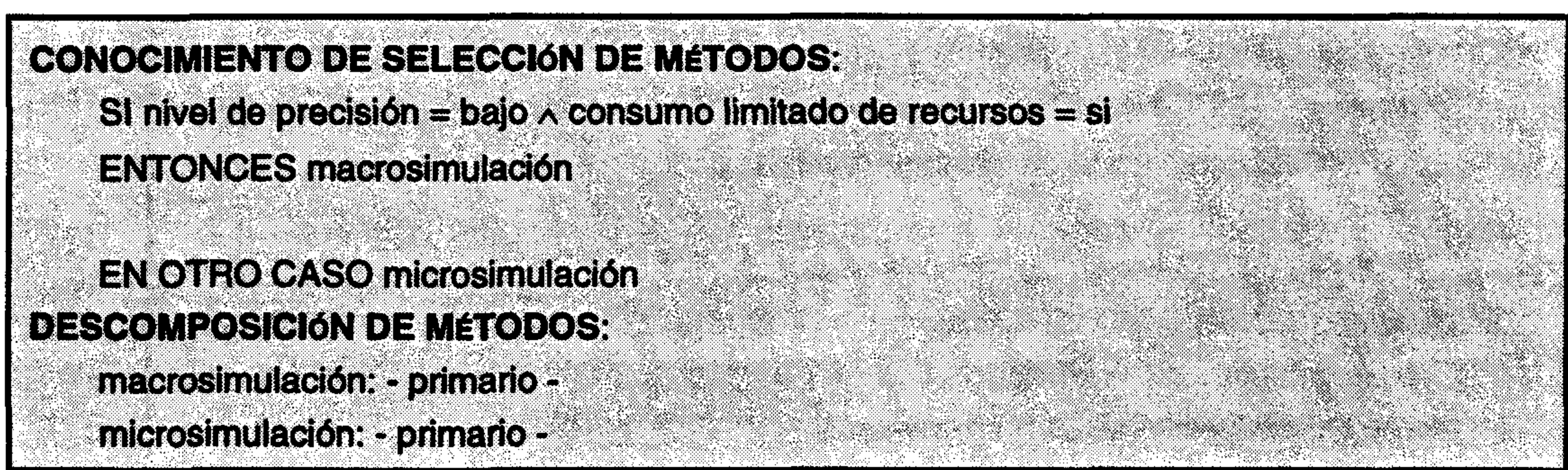


Figura 7.18: Especialista en la tarea de simular\_comportamiento

◆ Especialista en la tarea *clasificar situación*

La clasificación del estado de una línea se puede hacer de dos formas:

- (i) Con un método de *clasificación heurística*, o
- (ii) Con un método de *clasificación heurística con anticipación de eventos* que avisa además de sucesos previstos a corto plazo que podrían incidir en el estado de la línea.

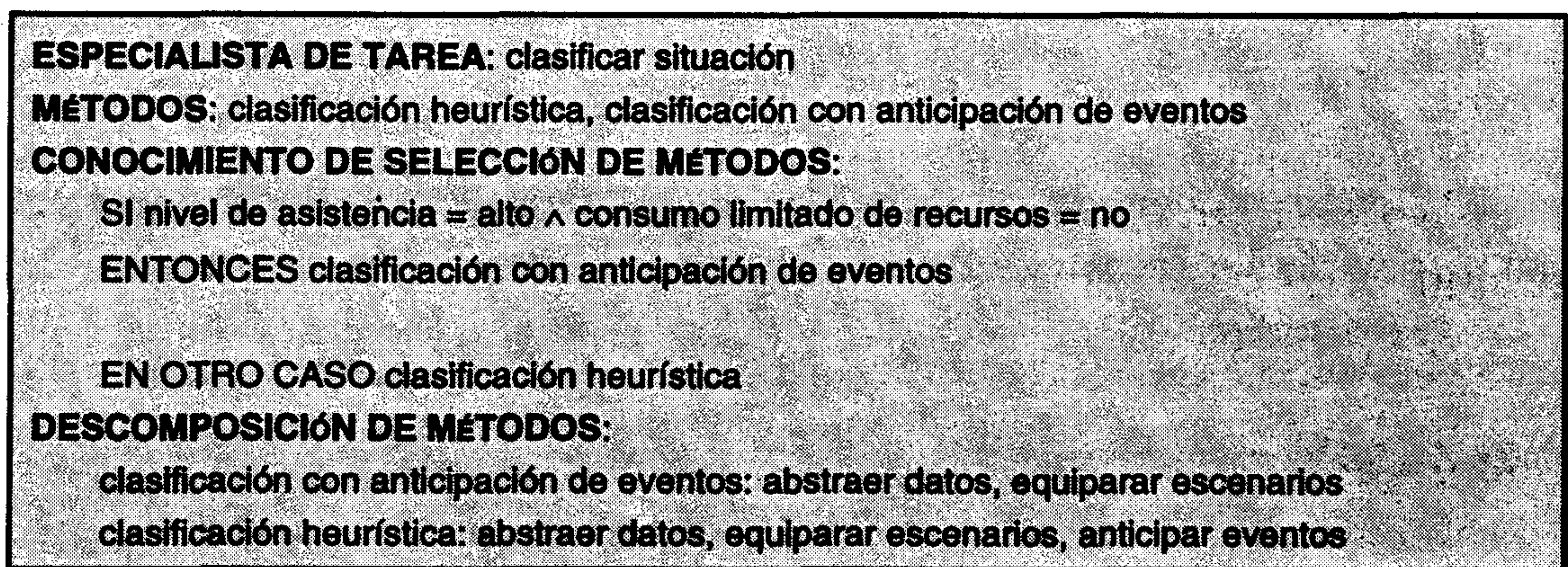


Figura 7.19: Especialista en la tarea de clasificar\_situación

La decisión entre ambos se toma como sigue:

- Si el nivel de asistencia requerido es alto o no hay restricciones de tiempo entonces se elige el método de *clasificación con anticipación de eventos*,
- En otro caso, se elige el de *clasificación heurística*.



## (2) Especialistas de tareas asociadas a preguntas del tipo *¿qué podría pasar?*

### ♦ Especialista en la tarea *predecir condiciones externas*

La predicción de las condiciones del entorno de una línea a corto plazo puede hacerse con dos métodos:

- (i) Un método matemático que calcula dichos parámetros por extrapolación, y
- (ii) Un método heurístico que aplica un proceso de equiparación sobre un grafo de dependencias temporales entre situaciones de una línea para deducir de forma aproximada cuáles serán sus condiciones futuras.

La decisión entre ambos se toma como sigue:

- Si se requiere que los datos utilizados sean recientes y que los datos generados sean precisos entonces se elige la *predicción por extrapolación*.
- En otro caso, se elige la *predicción por equiparación*.

**ESPECIALISTA DE TAREA:** predecir condiciones externas  
**MÉTODOS:** predicción por extrapolación, predicción por equiparación  
**CONOCIMIENTO DE SELECCIÓN DE MÉTODOS:**  
 Si madurez de la información = reciente v nivel de precisión = alto  
 ENTONCES predicción por extrapolación  
  
 EN OTRO CASO predicción por equiparación  
**DESCOMPOSICIÓN DE MÉTODOS:**  
 predicción por extrapolación: - primario -  
 predicción por equiparación: - primario -

Figura 7.20: Especialista en la tarea de *predecir condiciones externas*

## (3) Especialistas de tareas asociadas a preguntas del tipo *¿qué debería hacerse?*



◆ Especialista en la tarea *gestionar servicio*

Para decidir qué tipo de acciones de control se necesitan en una línea para mejorar su nivel de servicio pueden utilizarse cuatro métodos:

- (i) Un método que se limita a proponer un plan de control para la situación observada en una línea.
- (ii) Un método que además de elaborar un plan con las diferentes acciones a tomar avisa de eventos futuros que podrían incidir en el efecto esperado del plan.
- (iii) Un método que primero predice cuál será el estado de la línea a corto plazo y luego elabora un plan para mejorar su servicio a partir de esa información, y
- (iv) Un método completo que predice, planifica y avisa de eventos futuros.

La decisión entre cuál de ellos elegir se toma como sigue:

- Si el nivel de abstracción requerido para la respuesta es alto y el nivel de asistencia bajo entonces se elige el método de *planificación directa*.
- Si el nivel de abstracción es bajo y la situación observada en la línea se ha clasificado como un problema leve o como no problemática entonces también se elige el método de *planificación directa*.
- Si el nivel de abstracción es alto, el nivel de asistencia es alto y, la situación observada en la línea se ha clasificado como un problema importante entonces se elige el método de *planificación con anticipación de eventos*.
- Si el nivel de abstracción es bajo, el nivel de asistencia es alto y, la situación observada en la línea se ha clasificado como un



problema importante entonces se elige el método de *predicción, planificación & anticipación de eventos*.

- En otro caso, se elige el método de *predicción & planificación*.

**ESPECIALISTA DE TAREA:** gestionar servicio

**MÉTODOS:** planificación directa, planificación con anticipación de eventos, predicción & planificación, predicción, planificación & anticipación de eventos

**CONOCIMIENTO DE SELECCIÓN DE MÉTODOS:**

Si nivel de abstracción = alto  $\wedge$  nivel de asistencia = bajo  
**ENTONCES** planificación directa

Si nivel de abstracción = bajo  $\wedge$  pertenece(situación actual, problemas leves)  
**ENTONCES** planificación directa

Si nivel de abstracción = alto  $\wedge$  nivel de asistencia = alto  $\wedge$  pertenece(situación actual, problemas leves)  
**ENTONCES** planificación con anticipación de eventos

Si nivel de abstracción = bajo  $\wedge$  nivel de asistencia = alto  $\wedge$  pertenece(situación actual, problemas leves)  
**ENTONCES** predicción, planificación & anticipación de eventos

**EN OTRO CASO** predicción & planificación

**DESCOMPOSICIÓN DE MÉTODOS:**

planificación directa: decidir acciones de control

planificación con anticipación de eventos: decidir acciones de control, anticipar eventos

predicción & planificación: predecir estado, decidir acciones de control

predicción, planificación & anticipación de eventos: : predecir estado, decidir acciones de control, anticipar eventos

Figura 7.21: Especialista en la tarea de gestionar\_servicio

◆ Especialista en la tarea *decidir acciones de control*

La elaboración de un plan de acciones a tomar para mejorar el servicio de una línea puede hacerse con dos métodos:

- (i) Un método superficial que ofrece como respuesta dichos planes a nivel estratégico, y



- (ii) Un método de planificación jerárquica que proporciona un plan detallado de las acciones a tomar.

La decisión entre ambos se toma como sigue:

- Si el nivel de abstracción requerido en la respuesta es alto o el nivel de asistencia es bajo entonces se elige el método de *planificación superficial*,
- En otro caso, se elige el método de *planificación jerárquica*.

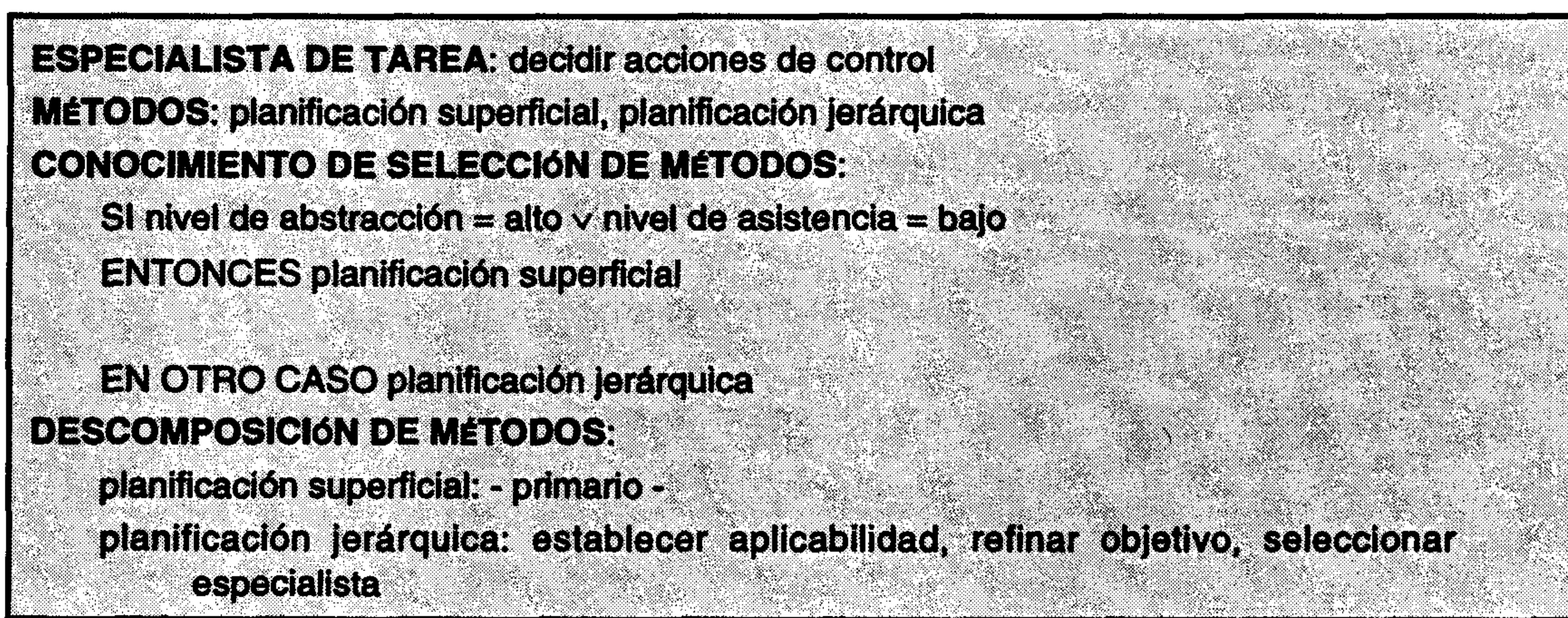


Figura 7.22: Especialista en la tarea de decidir acciones de control

## 7.6 Conclusiones

Este capítulo presenta la aplicación de la arquitectura descrita en el capítulo 6 al diseño de un sistema de ayuda a la decisión en la gestión del transporte público de la ciudad de Turín, integrado en el proyecto FLUIDS.

Este sistema proporciona información sobre el estado actual de las líneas de transporte público y sus vehículos, el estado previsto de las mismas bajo diferentes hipótesis sobre condiciones del entorno, y sobre las acciones de control que deberían llevarse a cabo para mejorar estados problemáticos tanto actuales como previsibles. Toda esta información se le ofrece al usuario a través de preguntas planteadas a lo largo de una conversación que pueden ser de tres tipos, atendiendo a las tres clases de información que el sistema puede ofrecer:



*¿qué está pasando?, ¿qué podría pasar? y ¿qué debería hacerse?*. La forma en la que se ofrecen las respuestas a cada una de las preguntas de estos tipos que puede formular un usuario se decide dinámicamente desde el Razonador Metanivel adaptándola a las características del usuario y sus preferencias con respecto al modo de interactuar con el sistema, así como a la disponibilidad de métodos de resolución de problemas capaces de satisfacer estos requisitos.





## 8. Implementación sobre KSM

---

Este capítulo presenta el prototipo del sistema de ayuda a la decisión, desarrollado en el proyecto FLUIDS [FLUIDS, 96], para la gestión de las líneas de autobuses de la ciudad de Turín. El diseño del modelo de conocimiento de este prototipo ya fue presentado en el capítulo 7 de acuerdo con las características de la arquitectura ROHCI propuesta en esta tesis, descrita en el capítulo 6.

En el prototipo se distinguen dos partes: (i) el interfaz de usuario multimedia, en el que cabe destacar un generador de presentaciones inteligentes multimedia, desarrollado por el DFKI (German Research Center for Artificial Intelligence) y basado en WIP (ver apartado 3.3.2), y (ii) el modelo de resolución de problemas, diseñado en base a la arquitectura ROHCI e implementado con la herramienta KSM. Al ser esta segunda parte la que sirve de apoyo a la validación del trabajo presentado en esta tesis, este capítulo está centrado en la descripción del modelo de resolución de problemas de transporte público construido con KSM. Aun así, al final del capítulo se presenta un ejemplo de operación en el que pueden verse algunos aspectos del interfaz de usuario del prototipo FLUIDS y, en el anexo, se incluye una descripción más detallada de este interfaz y en particular de la técnica de generación de presentaciones inteligentes utilizada.

Las facilidades de diseño y construcción de sistemas basados en el conocimiento que proporciona la herramienta KSM han dado lugar a un modelo genérico de gestión de los problemas asociados al transporte público que ha sido después particularizado para el dominio concreto de la red de transporte público de la ciudad de Turín. Como se recordará por la descripción del KSM incluida en el capítulo 2, KSM se presenta como una solución al diseño y construcción de sistemas basados en el conocimiento desde la perspectiva de una organización estática del conocimiento de resolución de problemas. Por tanto, la construcción del sistema de gestión de transporte público con el KSM supuso ampliar las funcionalidades de la herramienta para



que pudiese soportar modelos dinámicos de conocimiento, es decir, para que la forma de realizar una tarea pudiese decidirse en tiempo de ejecución eligiendo un método del conjunto de métodos de resolución de problemas disponibles.

El paradigma utilizado en KSM para la construcción, tanto del modelo genérico de gestión de transporte público como su instanciación en la ciudad de Turín, tiene como elemento principal de modelización la unidad cognitiva, que modeliza un área de conocimiento atribuible a un experto en un determinado campo. Cada modelo queda definido como una estructura jerárquica de unidades cognitivas en las que se distingue su conocimiento de su funcionalidad. Esto último hace posible que KSM ofrezca una visión funcional del sistema construido al mostrar la jerarquía de tareas y métodos asociada a la estructura de unidades cognitivas. La descripción de estos tres elementos: modelo genérico, modelo del dominio y modelo de tareas y métodos, son el objeto de este capítulo.

## 8.1 Modelo Genérico de Gestión de Transporte Público

De acuerdo con la organización modular de la metodología KSM empleada para desarrollar este modelo, todo el conocimiento de gestión de transporte público está representado por un área de conocimiento descompuesta en áreas más simples y especializadas, definiendo así una jerarquía. El modelo completo, denominado *Public Transport Management Model*, hace una primera división del conocimiento en dos grandes áreas:

- La relativa a todas las actividades de comunicación con el gestor del interfaz de usuario, modelizada en el área denominada *Dialogue Manager Interaction*. Aquí reside el conocimiento sobre:
  - (i) La traducción de las preguntas del usuario, expresadas en el lenguaje del gestor del interfaz, al lenguaje de resolución de problemas, es decir, a las tareas genéricas que deben ser resueltas para obtener las respuestas; y

- (ii) La traducción de las respuestas al formato apropiado para el gestor del interfaz.
- La relacionada con todas las actividades de resolución de problemas necesarias para generar las respuestas de la forma requerida por los usuarios, modelizada en el área llamada *Problem Solving Knowledge*.

El conocimiento de resolución de problemas se descompone en tres grandes clases:

- *Conocimiento clasificativo*. Necesario para identificar la situación de una línea a partir de un conjunto de datos de estado básicos enviados por el sistema de información. Está modelizado en el área de conocimiento *Classification Knowledge*.
- *Conocimiento predictivo*. Empleado para estimar la evolución a corto plazo de los vehículos, y a partir de ellos de las líneas, para evaluar la gravedad los problemas observados y proporcionar la información necesaria para corregir los retrasos de los vehículos involucrados. Está modelizado en el área de conocimiento *Prediction Knowledge*.
- *Conocimiento de planificación*. Usado para diseñar las acciones de control necesarias para resolver los problemas actuales de una línea o minimizar al menos sus consecuencias. Está modelizado en el área de conocimiento *Planning Knowledge*.

Estas tres áreas de conocimiento tienen su propio conocimiento reflexivo especializado con el que dinámicamente deciden la forma más apropiada de resolver el problema correspondiente, es decir, el método de resolución de problemas que ha de utilizarse, de acuerdo con las características de la conversación.

El área de conocimiento clasificativo divide a su vez su conocimiento de caracterización del estado de una línea en dos:

- Conocimiento para caracterizar el estado global de una línea como un valor de una escala de alarmas definido en el área de conocimiento *Alarms Level*.



- Conocimiento con el que proporcionar descripciones detalladas de los elementos que determinan el estado actual de la línea, como la distribución y valor medio del retraso entre los vehículos de una línea, la presencia de vehículos averiados u obstáculos en la línea, etc., definido en el área de conocimiento *Problem Scenarios*.

Además, una tercera partición del conocimiento del área *Classification Knowledge* es la correspondiente al conocimiento reflexivo con el que decidir dinámicamente el modo de resolver la tarea de clasificar el estado de una línea. Este conocimiento reflexivo se encuentra en el área de conocimiento denominada *Classification Reflective Knowledge*.

El área de conocimiento predictivo también divide su conocimiento en tres partes:

- Conocimiento con el que proporcionar una valoración global de la calidad del servicio a corto plazo en una línea, incluido en el área de conocimiento *Theoretic Prediction Model*.
- Conocimiento para proporcionar una descripción detallada del comportamiento esperado y la posición de los vehículos retrasados a corto plazo, incluido en el área de conocimiento *Behavior Model*.
- Conocimiento reflexivo con el que decidir la forma de hacer una predicción en función de las características de la conversación, incluido en el área de conocimiento *Prediction Reflective Knowledge*.

Por su parte, el conocimiento del área de conocimiento *Behavior Model* está dividido en cuatro partes:

- Conocimiento sobre los diferentes problemas que pueden presentarse en una línea con el paso del tiempo a partir de una situación de partida, incluido en el área de conocimiento *Problem Evolution Types*.
- Conocimiento sobre los distintos turnos de vehículos y conductores de una línea,, incluido en el área de conocimiento *Scheduled Behavior*.

- Conocimiento sobre los tiempos medios de recorrido de las distintas secciones de una línea, incluido en el área de conocimiento *Travel Times*.
- Conocimiento sobre los elementos componentes de una línea y su estructura, incluido en el área de conocimiento *Line Model*.

Con estas cuatro clases de conocimiento, el área *Behavior Model* sabe analizar si el servicio planificado difiere o no del que previsiblemente ofrezcan los vehículos actualmente con problemas a corto plazo, identificando así el estado futuro de la línea.

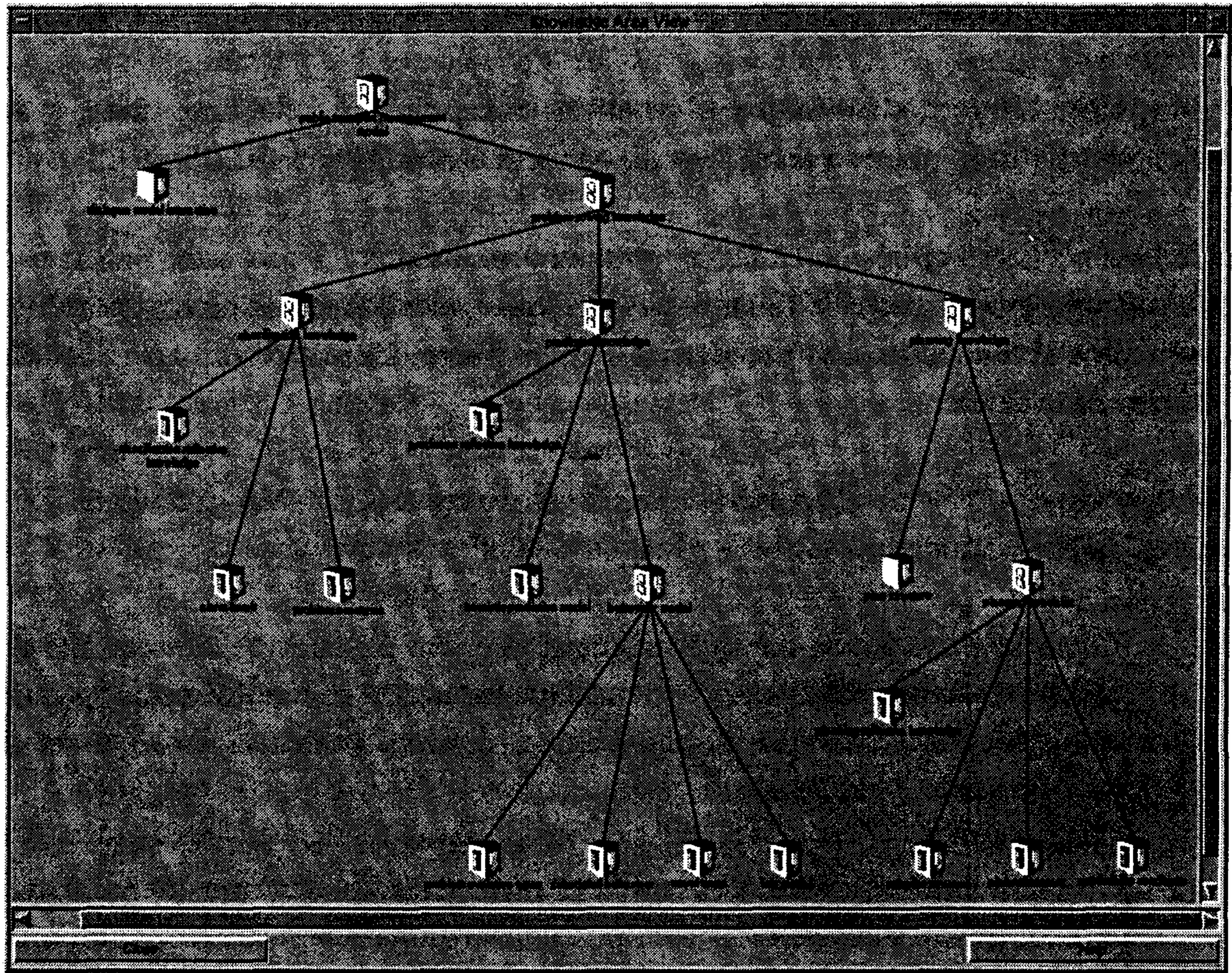
El conocimiento de planificación de acciones de control está dividido entre dos áreas de conocimiento:

- Conocimiento sobre cómo configurar un plan de control, decidiendo en cada uno de los pasos del plan qué tipo de acción es la requerida y quién debe proporcionar los detalles de la misma, incluido en el área de conocimiento *Plan Structure*.
- Conocimiento especializado en la planificación de acciones de control a diferentes niveles de abstracción, incluido en el área de conocimiento *Planning Specialist*.

Por su parte, el conocimiento de un especialista en planificación de acciones de control está dividido en:

- Conocimiento para decidir su aplicabilidad bajo las condiciones impuestas por el contexto del proceso de razonamiento, incluido en el área de conocimiento *Applicability Conditions*.
- Una vez establecida la aplicabilidad, conocimiento para seleccionar la clase de solución que debe aplicarse entre un conjunto de alternativas, incluido en el área de conocimiento *Subplan Structure*.
- Para cada plan alternativo, conocimiento sobre la descomposición en acciones atómicas y/o subplanes que definen el plan, incluido en el área de conocimiento *Plan Refinement*.





*Figura 8.1: Modelo de conocimiento genérico para la gestión del transporte público*

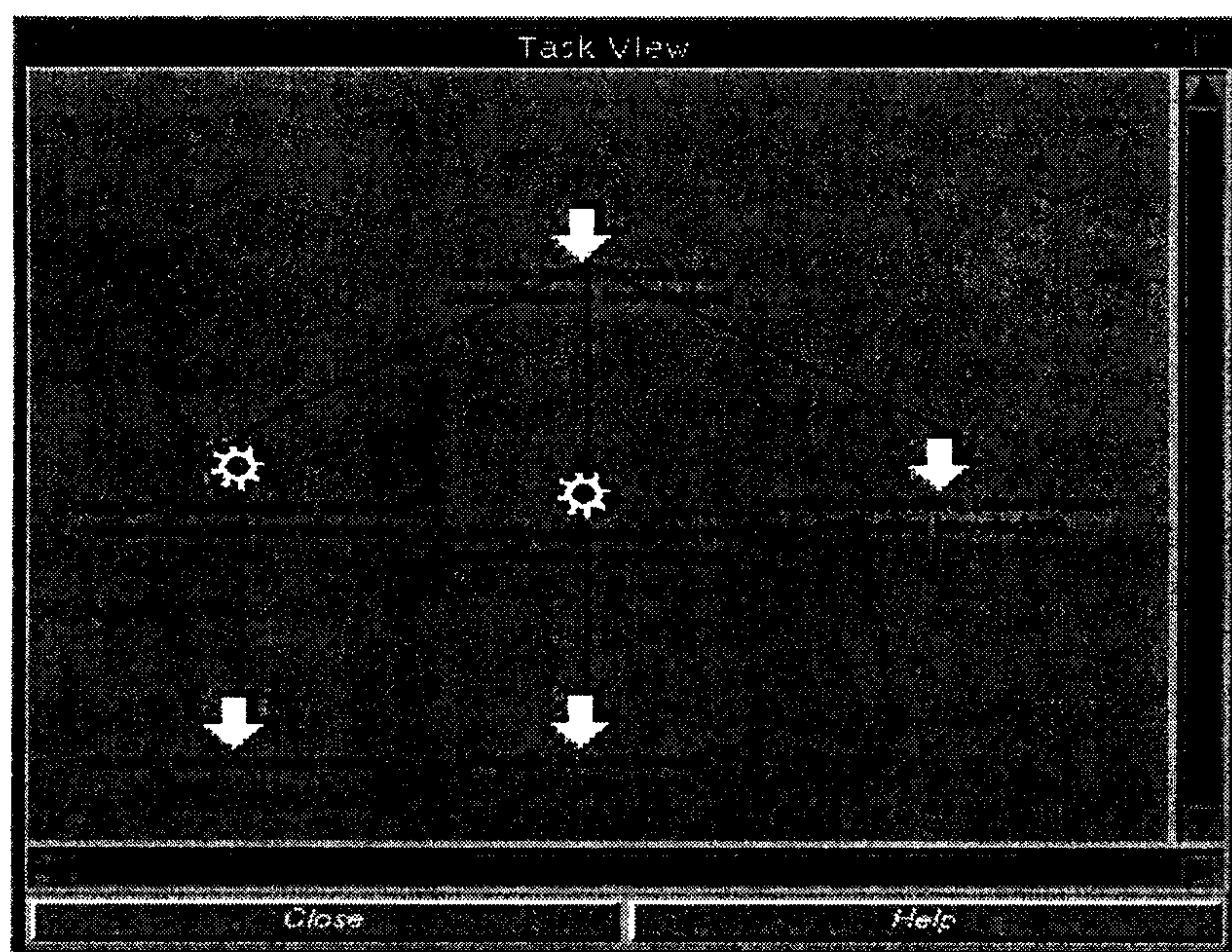
La organización de todas las diferentes áreas de conocimiento presentadas, representada como un modelo genérico de conocimiento KSM, se muestra en la figura 8.1. Este modelo genérico constituye una plantilla que define la forma de organizar el conocimiento y las estrategias generales de razonamiento aplicables a cualquier dominio en el que desee realizarse la gestión de una red de transporte público.

La instanciación de este modelo sobre un dominio concreto supone especificar el contenido de las distintas bases de conocimiento con la información del dominio, que en el caso de la aplicación mostrada en este capítulo se corresponde con parte de la red de transporte público de la ciudad de Turín.



## 8.2 Modelo de Tareas y Métodos

La visión funcional del modelo de conocimiento anterior presenta una jerarquía de tareas y métodos que definen las formas posibles de resolver los problemas asociados a la gestión del transporte público. La herramienta KSM ofrece la posibilidad de visualizar esta estructura jerárquica en la que las tareas están representadas por flechas y los métodos por tuercas. Cada tarea tiene como nodos descendientes en la jerarquía los métodos con los que puede llevarse a cabo, y por su parte, los descendientes de cada método son las subtareas en las que se descompone. Para simplificar la representación del modelo, si una tarea puede realizarse con un único método entonces dicho método no aparece gráficamente representado y se incluyen como descendientes de la tarea directamente las subtareas del método. Las tareas asociadas a métodos básicos constituyen el nivel inferior de la jerarquía.



*Figura 8.2: Parte de la jerarquía de tareas-métodos centrada en la funcionalidad de clasificación*

La estructura de tareas y métodos asociada al modelo genérico descrito anteriormente es la siguiente:

- La clasificación del estado de una línea se realiza mediante una tarea de identificación de escenarios problema que empieza por hacer una abstracción de los datos básicos enviados por el sistema de información



para después hacer la identificación utilizando dos tipos de métodos basados en reglas (ver figura 8.2):

- (i) Uno superficial que a partir de información de estado deduce niveles de alarma sobre el estado de la línea, y
- (ii) Uno más completo que especifica los elementos que inciden en el estado de la línea.

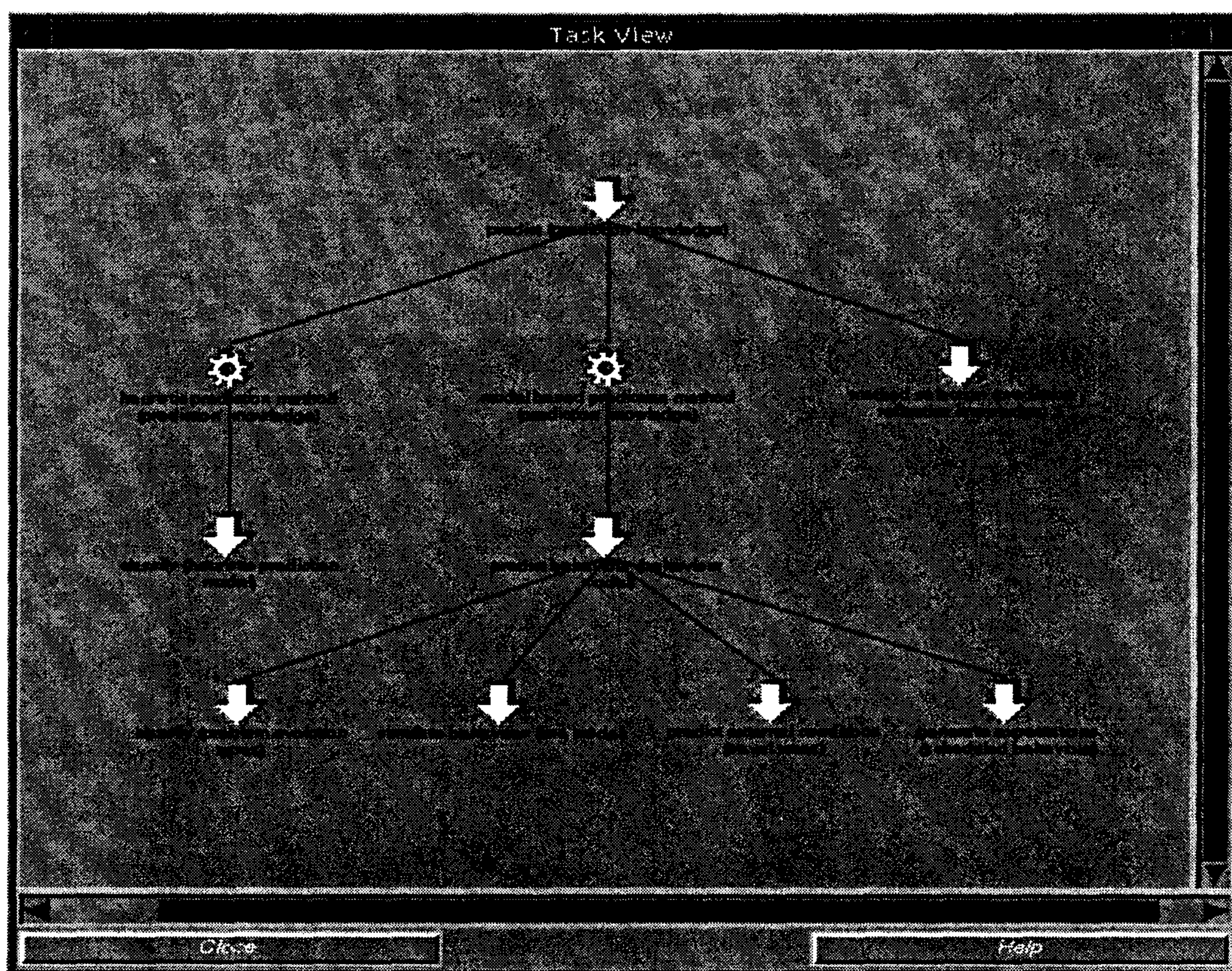


Figura 8.3: Parte de la jerarquía de tareas-métodos centrada en la funcionalidad de predicción

- La tarea de predicción se puede realizar también con dos métodos (ver figura 8.3):
  - (i) Uno clasificativo que utiliza un modelo de predicción heurística, definido a partir de la experiencia de los operadores, que asocia la presencia de eventos relevantes observados con estados futuros previsibles; y



- (ii) Uno más complejo en el que se llevan a cabo cuatro subtareas:
  - (1) Estimación de los tiempos medios de recorrido de los distintos sectores de la línea.
  - (2) Simulación de la evolución a corto plazo de los vehículos retrasados.
  - (3) Extracción de una base de datos del horario previsto para los vehículos retrasados.
  - (4) Comparación de los estados esperado y previsible de los vehículos retrasados para inferir posibles cortes en el servicio y la gravedad de los mismos.
- La planificación de acciones de control, a su vez, se puede hacer también de dos formas (ver figura 8.4):
  - (i) Con un método superficial que proporciona las líneas generales de los planes de control aplicables y que está soportado por una base de conocimiento de reglas que relacionan posibles escenarios problemáticos de línea con planes de control a alto nivel.
  - (ii) Con un método detallado que emplea la estrategia de *establecer&refinar* con un conjunto de especialistas en la configuración de diferentes aspectos de un plan de control. Este método hace una búsqueda jerárquica basada en un bucle de tres etapas, llevado a cabo por subtareas más básicas, hasta la especificación final del plan de control. Las subtareas asociadas a este método son tres:
    - (1) Primero, el especialista analiza si la entrada satisface las condiciones de aplicabilidad del especialista. Inicialmente, se selecciona el especialista de más alto nivel en la jerarquía y sus condiciones de aplicabilidad se consideran satisfechas por defecto.
    - (2) Entonces, el conocimiento de refino del especialista se aplica para identificar cuál de las posibles soluciones/acciones de control de que dispone es la más adecuada para ser adoptada.



- (3) Finalmente, se analizan las distintas etapas de la solución elegida anteriormente para identificar aquellas cuya realización requiere la intervención de otros especialistas.

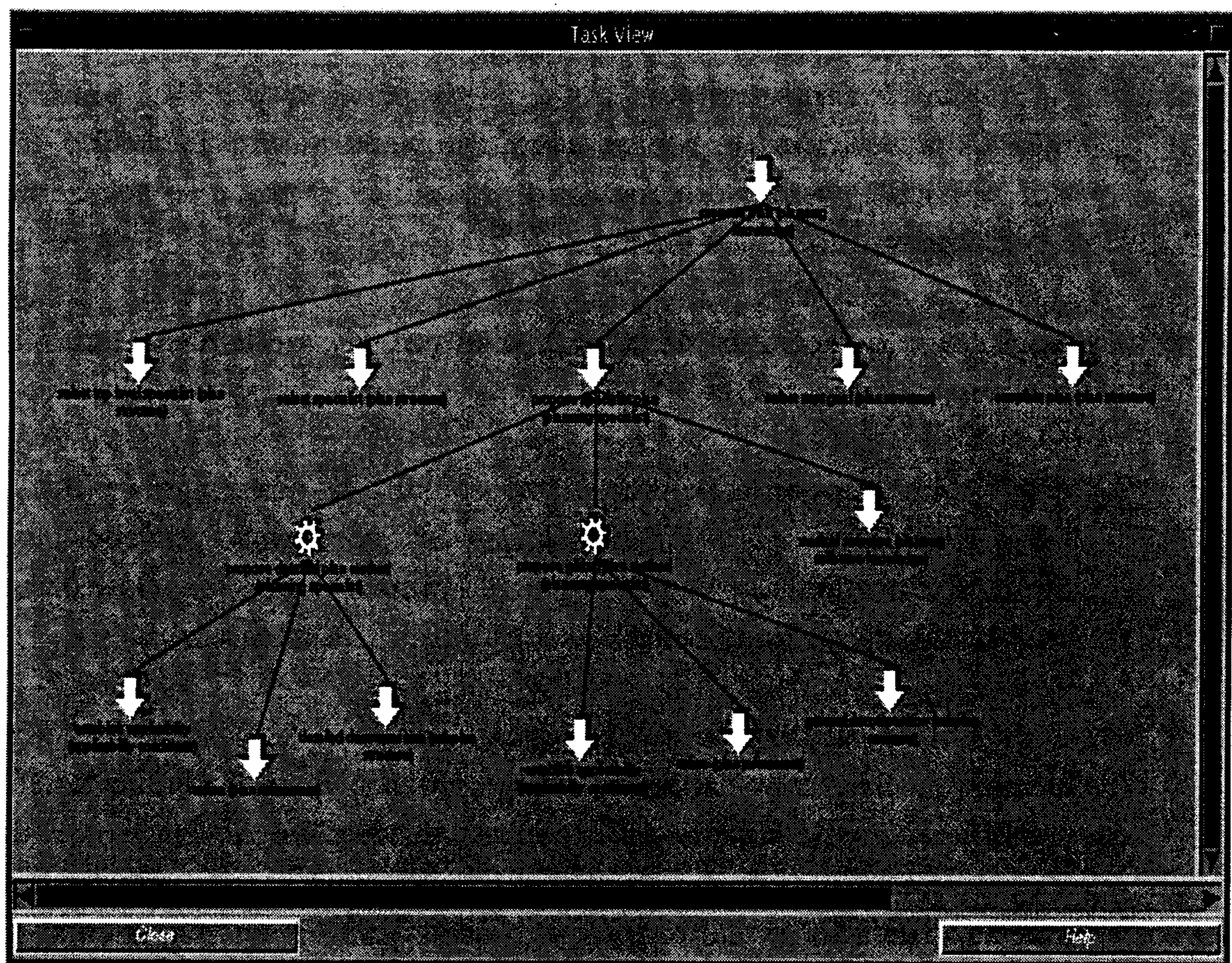


Figura 8.4: Parte de la jerarquía de tareas-métodos centrada en la funcionalidad de planificación

El conocimiento reflexivo empleado para decidir dinámicamente la estructura de tareas adecuada para resolver cada clase de problema, está distribuido entre las tres actividades principales, es decir, clasificación, predicción y planificación. El razonamiento se lleva a cabo mediante una tarea de selección soportada por reglas que relacionan atributos de interacción con el correspondiente método de resolución de problemas.

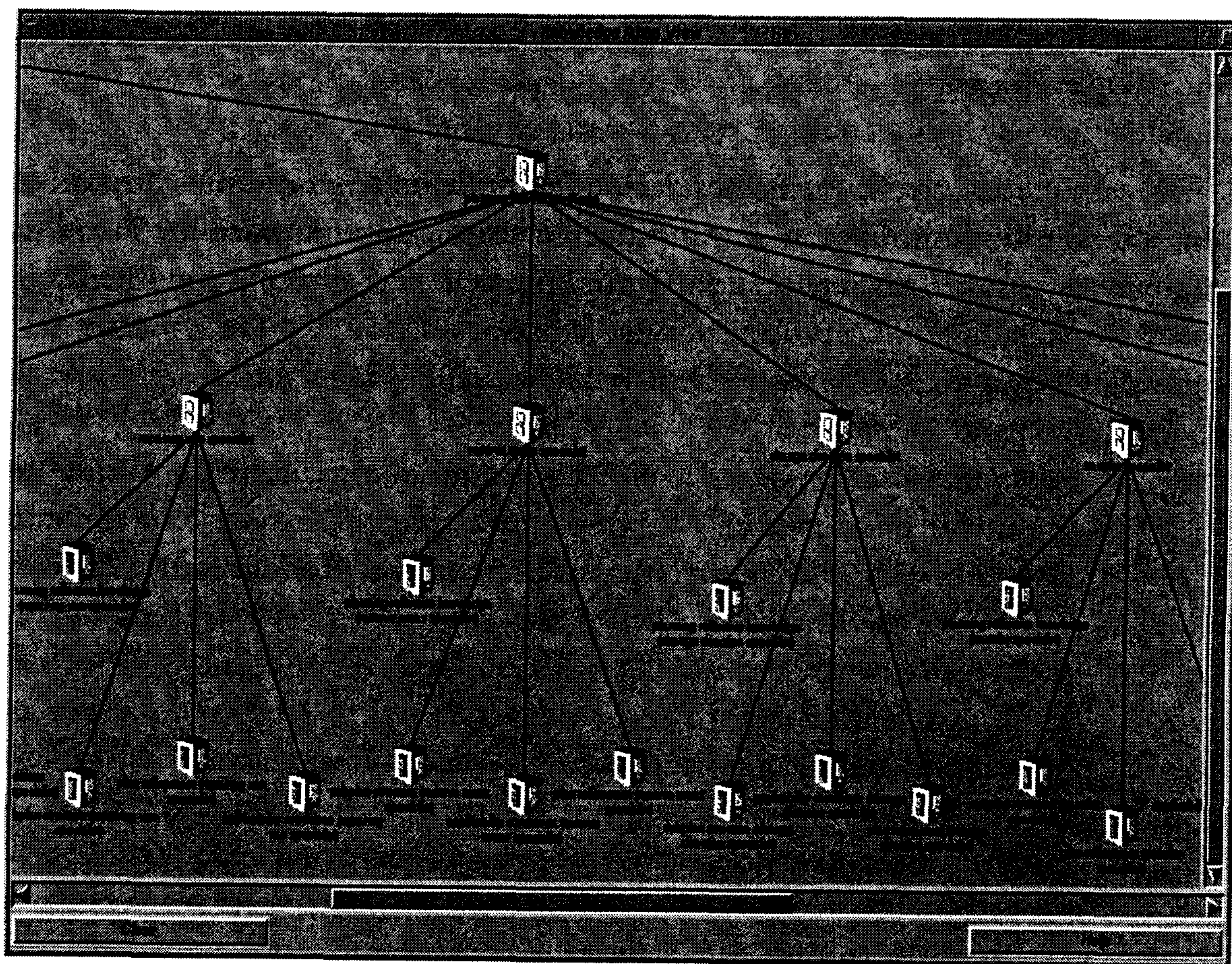


## 8.3 Modelo del Dominio de la Red de Autobuses de Turín

Esta sección describe la particularización del modelo genérico antes descrito en un modelo de dominio con la información propia de una parte de la red de transporte público de la ciudad de Turín. En particular, el prototipo FLUIDS se ha desarrollado con la información de las líneas 4, 14 y 63. Los aspectos más relevantes de este proceso de particularización son los siguientes:

- ♦ *Instanciación de unidades cognitivas* del modelo genérico en unidades cognitivas del dominio. Así por ejemplo, para el área de conocimiento *Planning Knowledge* se crea una instancia denominada *Planning Knowledge for line 63 in Turin* que sabe cómo elaborar planes de acciones de control para la línea 63 de la red de transporte público de Turín. Este área *Planning Knowledge for line 63 in Turin* además hereda del modelo genérico los tipos de áreas de conocimiento que va a poder utilizar, lo que implica que el proceso de instanciación continúe con el resto de áreas de conocimiento que vayan apareciendo. Obviamente, el proceso de instanciación empieza por el área de conocimiento de más alto nivel, es decir, *Public Transport Management Model*.
- ♦ *Duplicación de unidades cognitivas*. Además de la creación de las unidades cognitivas del dominio asociadas a la estructura planteada por el modelo genérico, que indica la herramienta KSM de forma automática, existen una serie de unidades cognitivas a nivel genérico que pueden dar lugar a más de una instancia en un modelo de dominio. Por ejemplo, la unidad *Planning Specialist* y todas las unidades en las que se divide son duplicadas seis veces para cubrir el número de especialistas en distintos aspectos de la gestión del transporte público en Turín: selección de rutas alternativas que esquiven secciones de línea bloqueadas, transferencia de pasajeros entre vehículos, intercambio de conductores, selección de vehículos de reserva, selección de vehículos de reparaciones, y cambio de dirección de vehículos. La figura 8.5 muestra la parte del modelo del dominio de Turín donde pueden apreciarse alguno de estos especialistas.





*Figura 8.5: Visión parcial del modelo de dominio para Turín*

- ◆ *Definición de conceptos y bases de conocimiento.* Los pasos anteriores producen la definición de la estructura completa de áreas de conocimiento que componen el modelo del dominio para la ciudad de Turín. En este punto se definen los vocabularios conceptuales asociados a las áreas de conocimiento primarias y el contenido de las bases de conocimiento del modelo.

Cada una de las áreas de conocimiento primarias definidas a nivel genérico tiene asociado un vocabulario conceptual genérico, en el que se establece una terminología de uso común a varias áreas, que en el caso del dominio de transporte público incluye todos los conceptos relativos a la topología de las líneas de autobuses, tales como paradas, puntos de cambio, áreas de reserva o depósitos. Para cada uno de estos conceptos se define una descripción genérica formulada en términos de clases con conjuntos de atributos y facetas junto con sus respectivos



dominios que es usada posteriormente para caracterizar los elementos de cada línea particular como un conjunto de instancias de las clases anteriores. La figura 8.6 muestra una ventana KSM en cuya mitad superior puede observarse parte de la descripción genérica de la clase parada y en cuya mitad inferior aparecen varias paradas concretas, resultado de instanciar dicha clase sobre la línea 63.

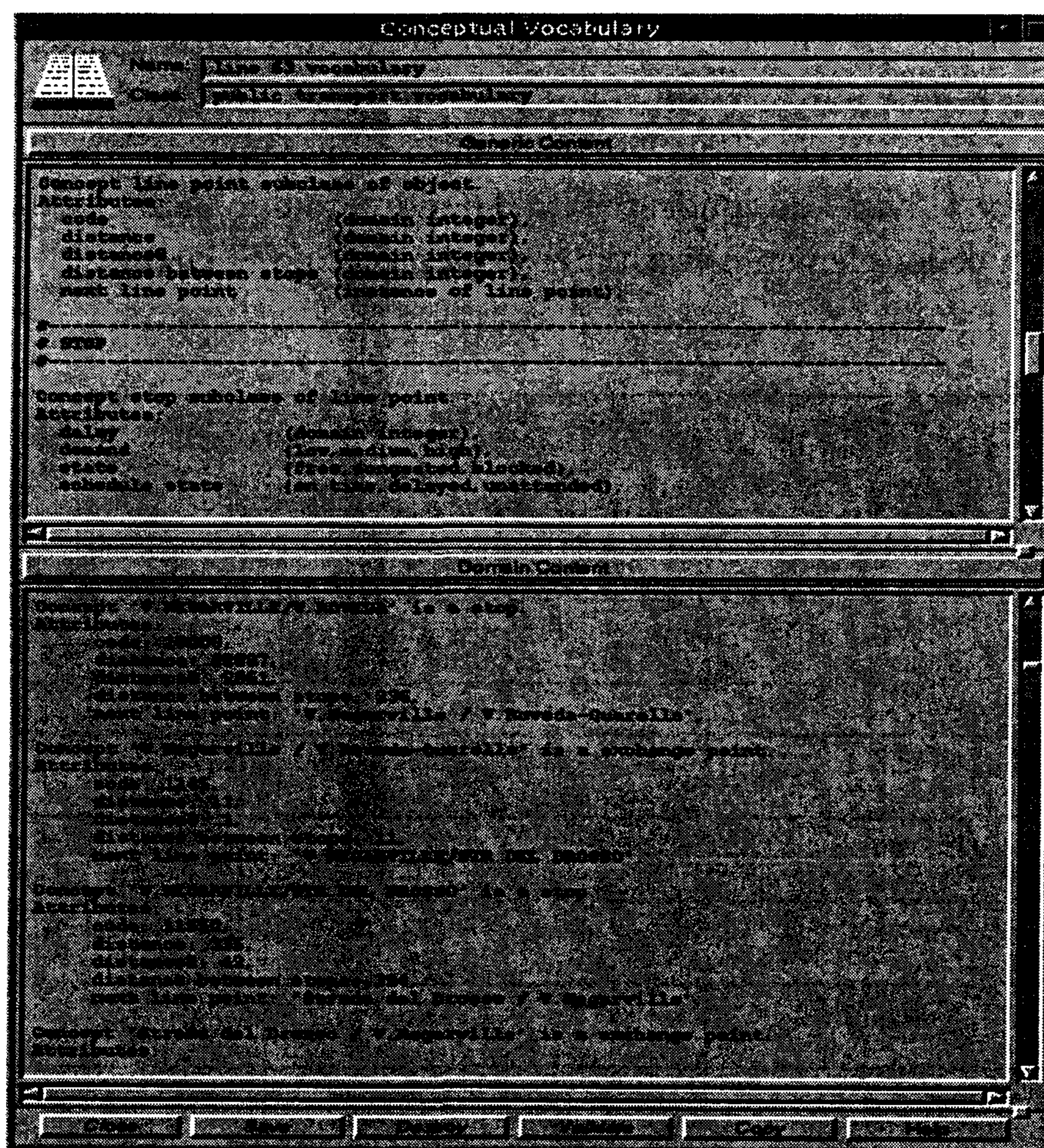


Figura 8.6: Ejemplo de vocabulario conceptual

Los vocabularios conceptuales del dominio son utilizados para generar las instancias de las bases de conocimiento declaradas en el modelo genérico con el conocimiento propio del área problema de Turín. La figura 8.8 muestra parte de las bases de conocimiento utilizadas para realizar las tareas de clasificación y predicción sobre la línea 63, que incluyen: tiempos de recorrido de los sectores de la línea agrupados por franjas horarias, modelo de conceptos agregados de la línea,



horario previsto de los vehículos y tipos de problemas que pueden presentarse en la línea. Cada una de estas bases tiene su propio modelo de representación del conocimiento. Por ejemplo, los tipos de problemas están descritos mediante reglas, mientras que los horarios se representan con tablas y los tiempos de recorrido con tablas que incluyen referencias temporales.

El modelo genérico incluye un total de 11 tipos de bases de conocimiento que emplean diferentes formalismos de representación: 3 de ellas relativas al conocimiento reflexivo y 9 con conocimiento de resolución de problemas de transporte público. El modelo del dominio incluye un total de 27 bases del primer tipo y 71 bases de conocimiento del segundo tipo. La figura 8.7 muestra un resumen de los elementos incluidos en el prototipo FLUIDS.

Line Model	concepto-atributo instancia de clases	147 conceptos 691 atributos	138 conceptos 651 atributos	110 conceptos 491 atributos	11 conceptos 45 atributos
Problem Schemas	reglas	—	—	—	20 reglas
Alarm levels	reglas	—	—	—	6 reglas
Scheduled Behaviors	tablas	188 celdas en 13 horarios	259 celdas en 20 horarios	155 celdas en 9 horarios	—
Historical Time Time	marcos	80 campos	80 campos	40 campos	—
Historical Prediction Model	reglas	—	—	—	12 reglas
Plan Structure (7 bases)	reglas	3 reglas	2 reglas	5 reglas	7 reglas
Plan Structure (7 bases)	reglas	217 reglas	187 reglas	161 reglas	7 reglas (47 condiciones)
Plan Structure (7 bases)	reglas no-good	3 reglas	2 reglas	3 reglas	—
Plan Structure (7 bases)	reglas	—	—	—	23 reglas
Plan Structure (7 bases)	reglas	—	—	—	11
Plan Structure (7 bases)	reglas	—	—	—	23 reglas

Figura 8.7: Resumen cuantitativo de los elementos del prototipo FLUIDS



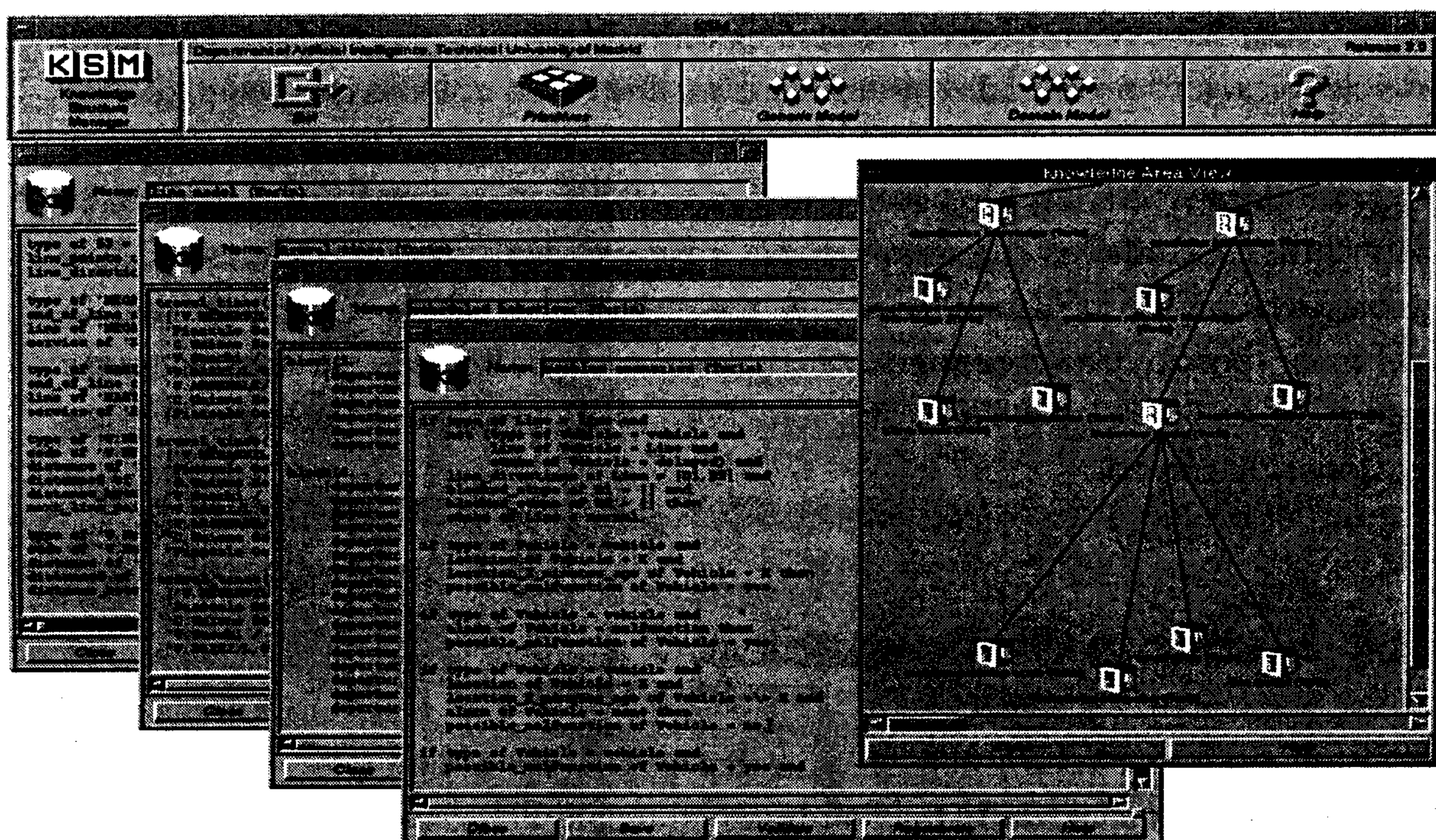


Figura 8.8: Bases de conocimiento para clasificación y predicción de problemas

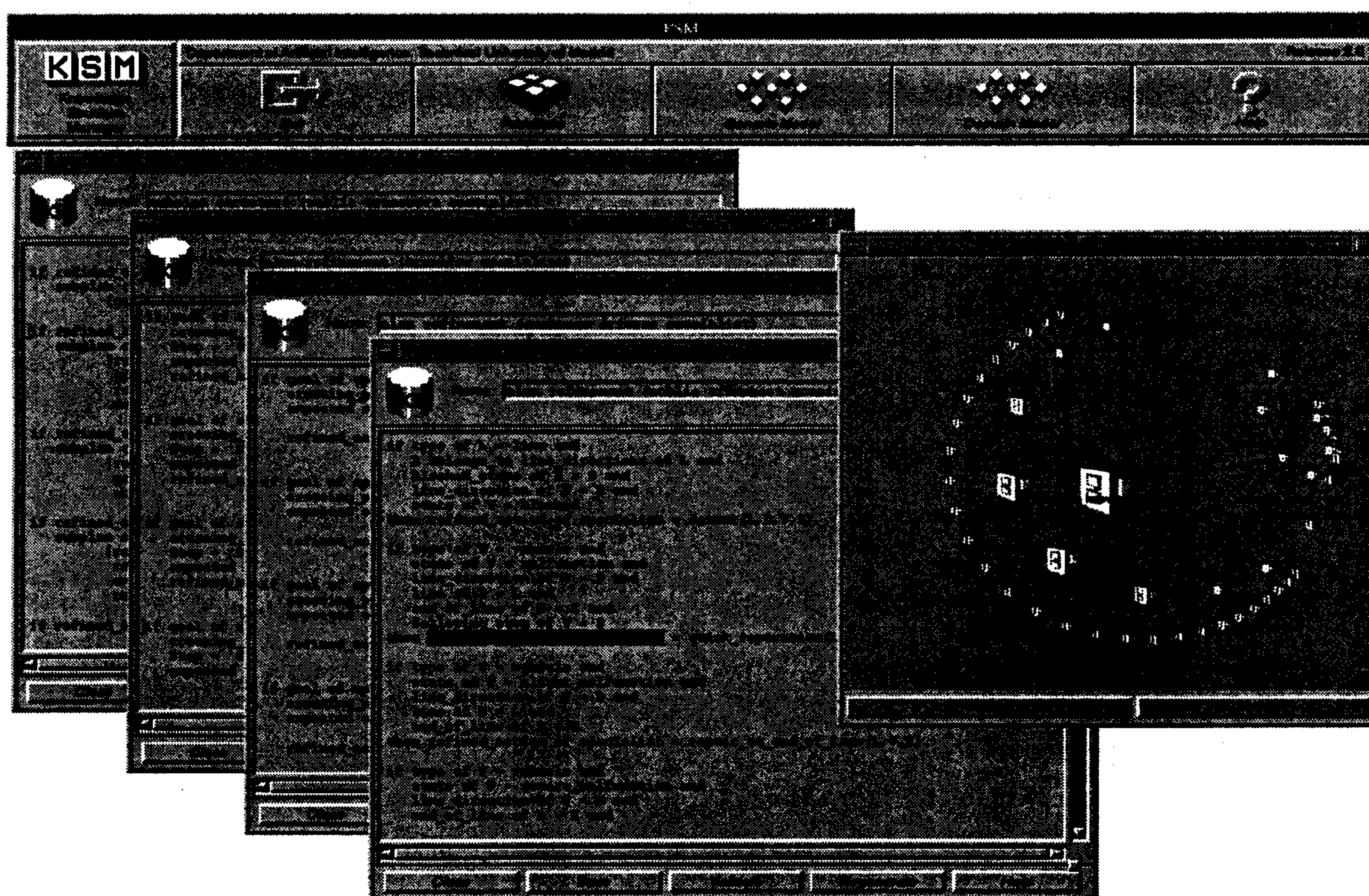


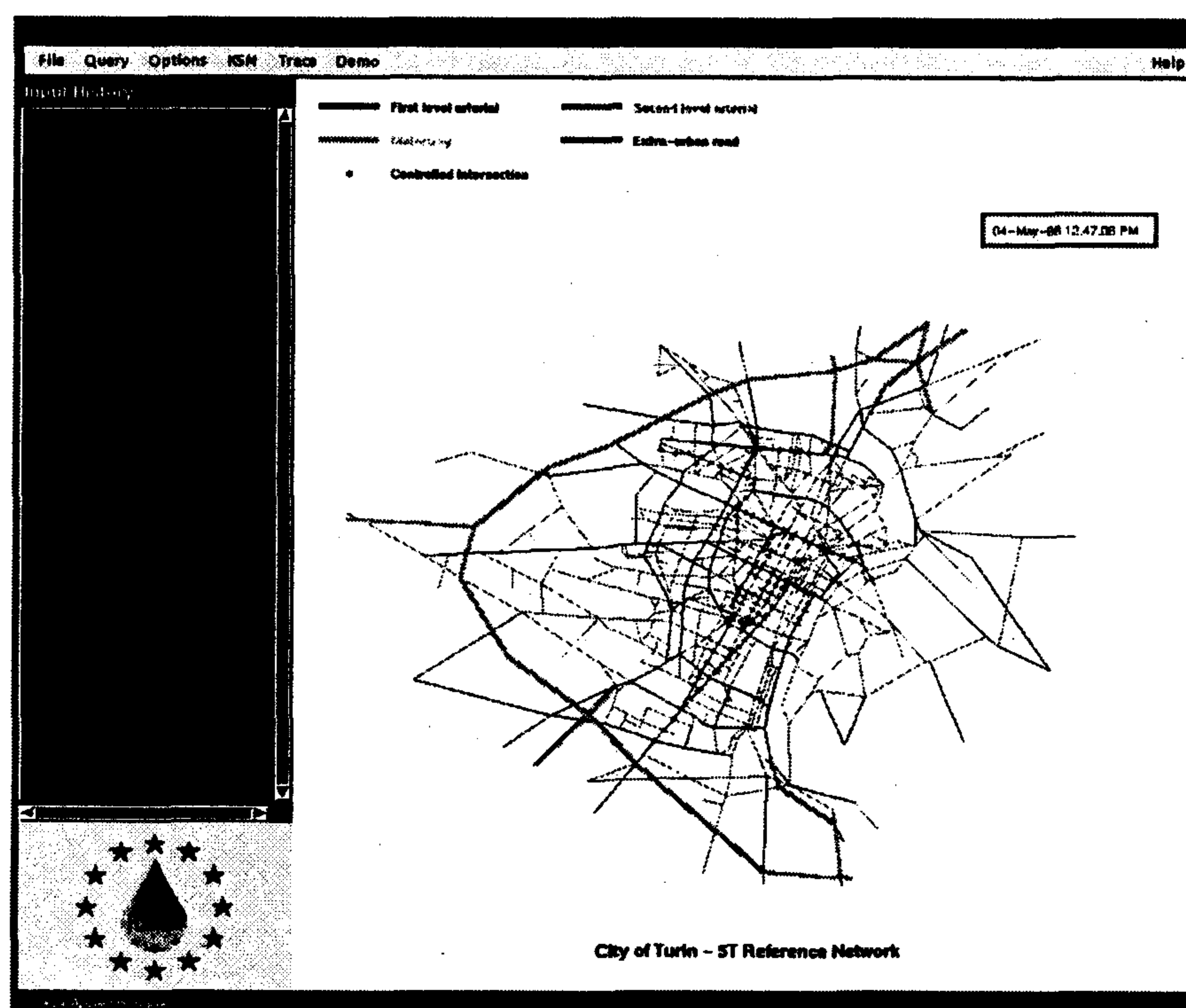
Figura 8.9: Perspectiva hiperbólica del modelo de dominio para Turín



Las figuras 8.8 y 8.9 muestran parte del contenido de las bases de conocimiento de clasificación, predicción y planificación, e ilustran además las posibilidades de representación de la estructura de modelos de conocimiento que ofrece la herramienta KSM. En el caso de la figura 8.8 puede verse una parte de la jerarquía de unidades cognitivas donde reside el conocimiento de clasificación y predicción de problemas, mientras que en la figura 8.9, la visión del modelo se hace desde una perspectiva hiperbólica en la que se incluye una representación del modelo completo pero focalizada en el planificador de acciones de control.

## 8.4 Ejemplo de Operación

Aunque los aspectos propios del interfaz de usuario no han sido el centro de atención de esta tesis, dado que la implementación de la arquitectura propuesta se ha llevado a cabo en el marco del proyecto FLUIDS y que uno de los pilares de este proyecto es la aplicación de interfaces inteligentes multimedia; este apartado presenta un ejemplo de la operación llevada a cabo por el prototipo FLUIDS desde la perspectiva del interfaz de usuario.



*Figura 8.10: Ventana inicial del prototipo FLUIDS para Turín*

El interfaz multimedia de FLUIDS soporta entradas y salidas multimodo. Con respecto a las entradas, se combinan preguntas formuladas en lenguaje natural y soportadas por menús con gestos deícticos con los que se seleccionan áreas potencialmente problemáticas como argumentos opcionales de las preguntas del tipo *¿qué está pasando?*. La presentación de la salidas puede configurarse utilizando varios medios: texto en lenguaje natural, sonido, representaciones gráficas, como mapas y diagramas informativos, y animación.

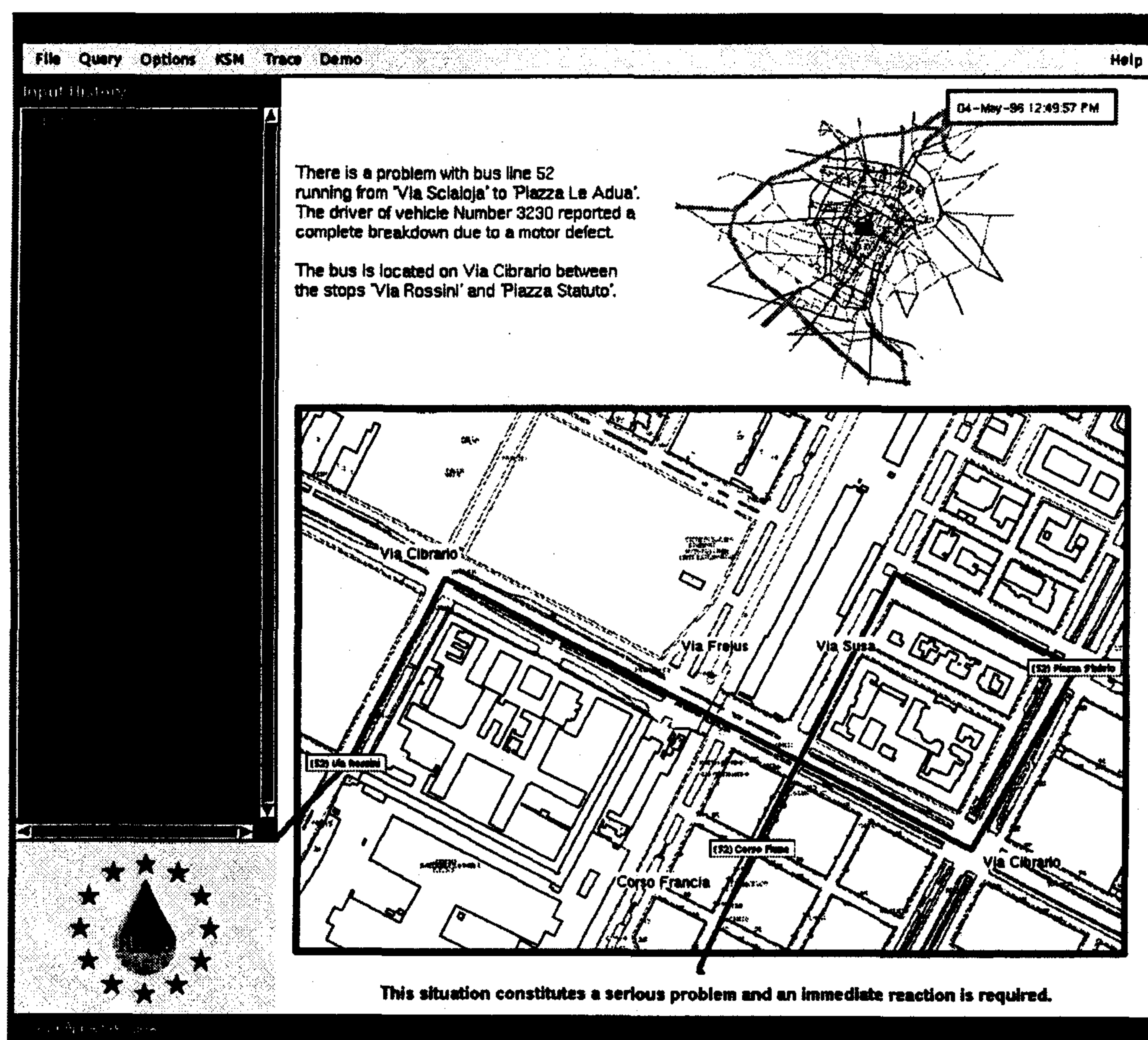


Figura 8.11: Ventana FLUIDS presentando el estado de una línea

La figura 8.10 muestra la ventana inicial del prototipo del proyecto FLUIDS sobre la gestión de la red de transporte público en la ciudad de Turín. A través de esta ventana se puede acceder a un menú en el que aparecen las preguntas que es posible formular, que dependen del contexto en el que se planteen, y al modelo de conocimiento soportado por el KSM. Cuando el sistema detecta algún problema en alguna de las líneas, hace un zoom sobre el mapa general



del área y muestra una representación detallada de la línea junto con la descripción de la situación detectada. Esa representación puede ser (i) un esquema de la línea completa, en la que se representan las paradas, la posición de los vehículos de servicio y diferentes códigos de colores con los que se indica el estatus de esos vehículos, o bien (ii) puede ser un mapa detallado de una parte de la línea, como se muestra en la figura 8.11.

Una vez descrito el problema de la línea, por ejemplo el del bloqueo de la línea 52 debido a la avería del vehículo 3230 de la figura 8.11, el usuario podría consultar por la evolución de la situación de la línea, lo que generaría una respuesta similar a la de la figura 8.12, en la que mediante códigos de color asociados a las barras verticales y el trazado de la línea sobre el mapa se representaría el nivel de servicio de la zona de la línea con problemas.

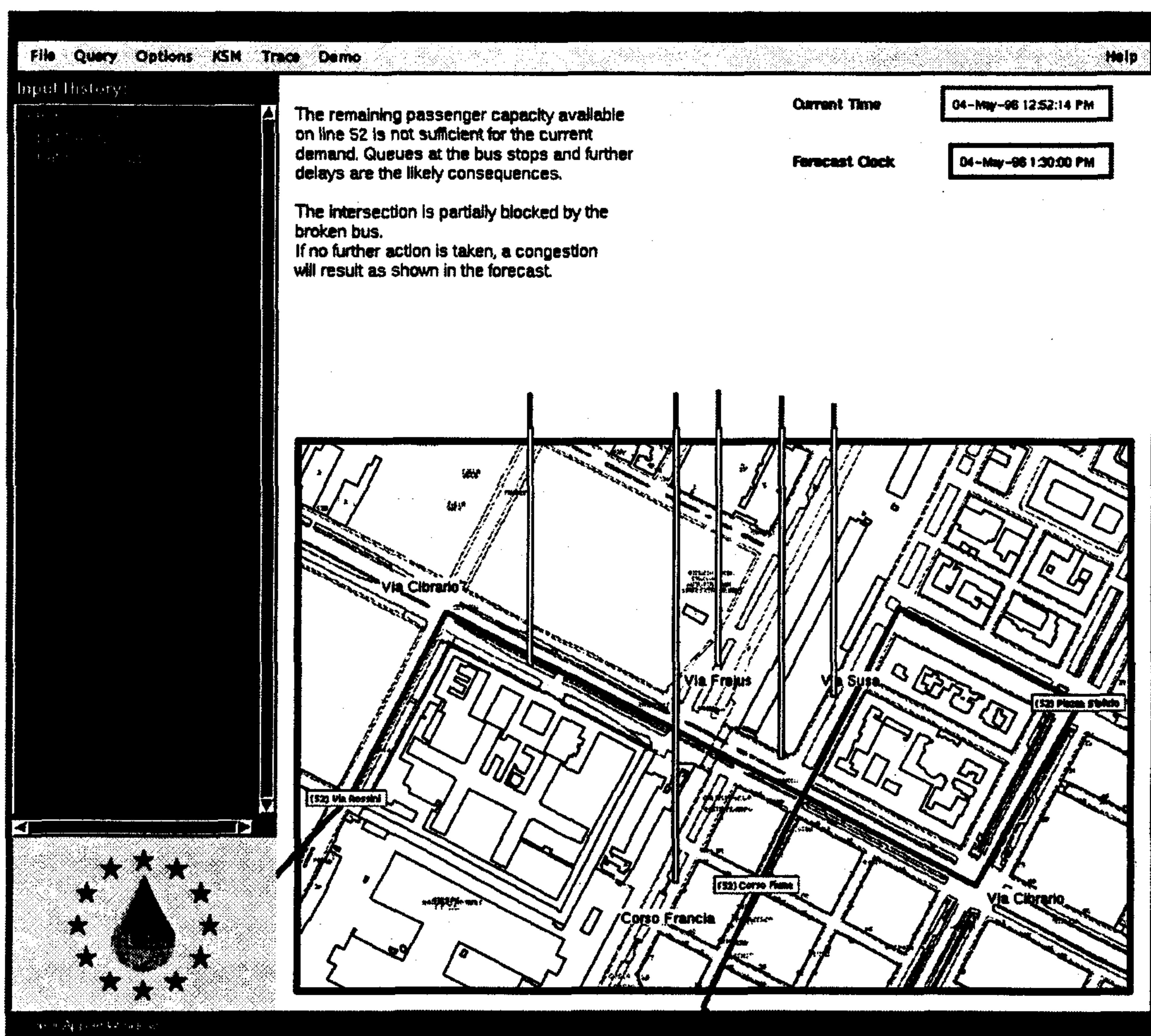


Figura 8.12: Ventana FLUIDS con la descripción del estado de una línea a corto plazo

La selección de la pregunta *what can be done?* en el menú de preguntas hace que se genere una propuesta de acción de control que en este ejemplo consiste en seleccionar un vehículo de reserva con el que reemplazar el vehículo averiado ya que éste no puede desplazarse hasta la cabecera de línea. El vehículo de reserva elegido será aquel, de entre los próximos al vehículo siniestrado, que esté en condiciones de incorporarse a la línea 52 recuperando el horario previsto para el vehículo 3230. La ventana de la figura 8.13 muestra esta respuesta en forma textual, representando los vehículos de reserva disponibles y moviendo en la dirección del vehículo averiado el vehículo de reserva seleccionado. Al pedir una justificación a la propuesta de control presentada en la figura 8.13 se obtiene la descripción del criterio empleado para seleccionar el vehículo de reserva, tal como se muestra en la figura 8.14.

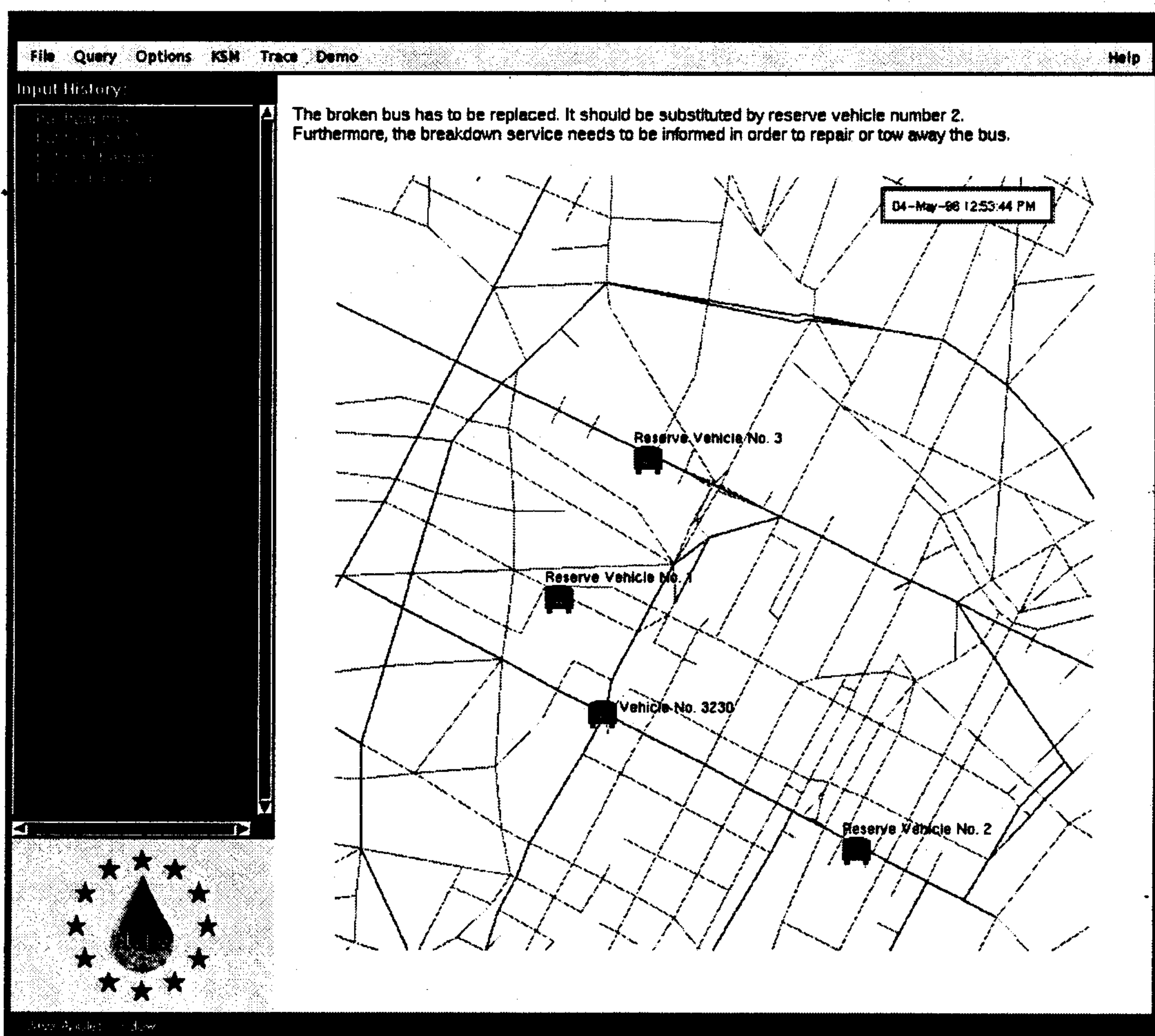


Figura 8.13: Ventana FLUIDS con la descripción de una propuesta de acción de control



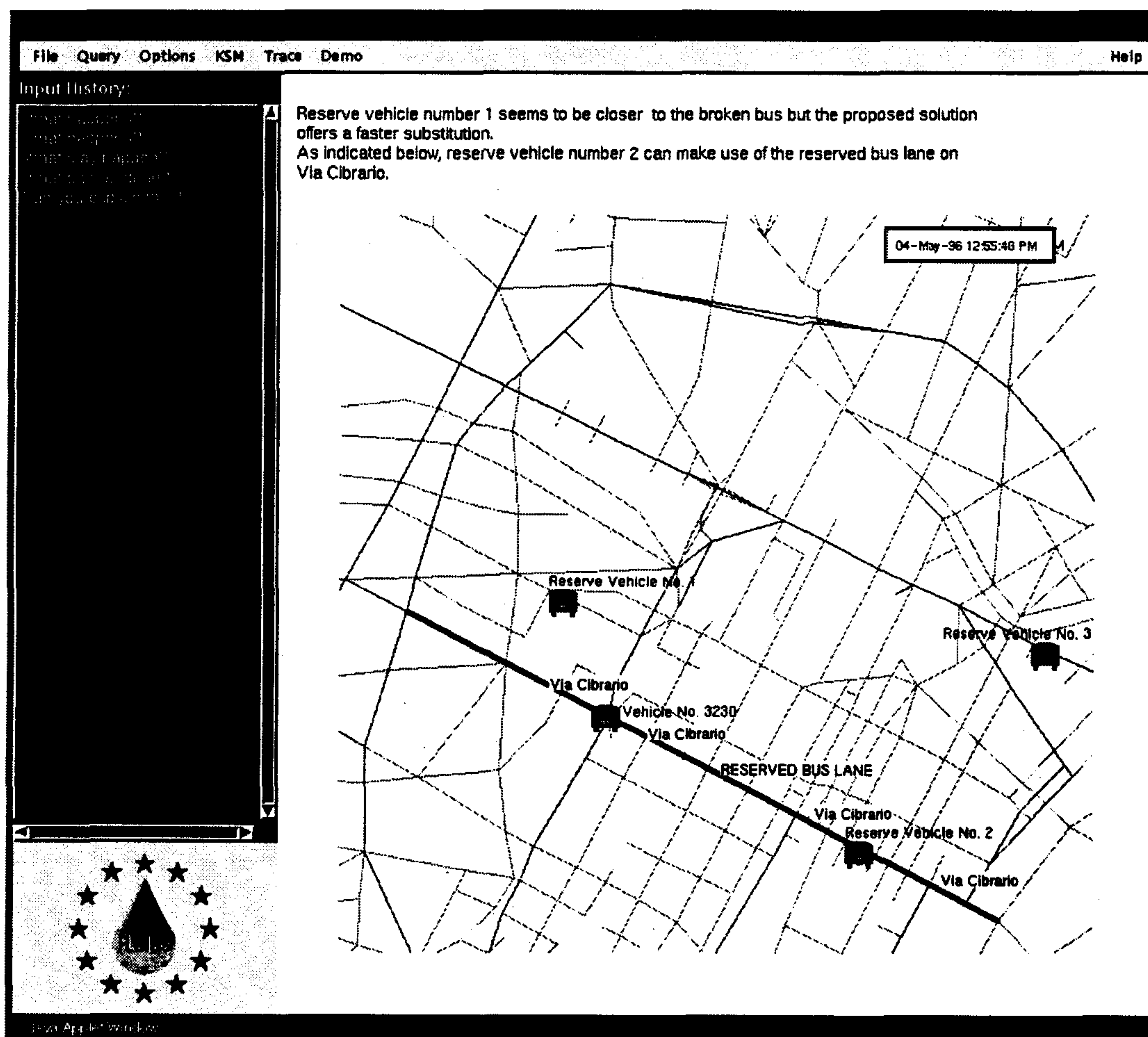


Figura 8.14: Ventana FLUIDS con la explicación de una propuesta de acción de control

Por otra parte, además de las herramientas gráficas disponibles para visualizar las estructuras de conocimiento de un modelo de resolución de problemas, como las que se han mostrado en las secciones 8.1, 8.2 y 8.3, la herramienta KSM también ofrece la posibilidad de monitorizar la ejecución del modelo, mostrando paso a paso la evolución del proceso de razonamiento. Esta funcionalidad emplea una representación del modelo de conocimiento genérico en la que, en función del estado del razonamiento, se colocan sobre las unidades cognitivas correspondientes etiquetas coloreadas con el nombre de la tarea que está siendo realizada. Estas etiquetas tienen cuatro significados distintos:

- *Verde*. Representa que una tarea ha sido llamada.
- *Blanco*. Representa que se ha encontrado una solución.



- *Amarillo*. Representa la repetición de la llamada a una tarea para producir nuevos resultados siguiendo una búsqueda no determinista.
- *Rojo*. Representa un fallo en la ejecución del método, es decir, que no se ha encontrado ninguna solución.

La figura 8.15 muestra un momento de la ejecución del modelo de gestión del transporte público.

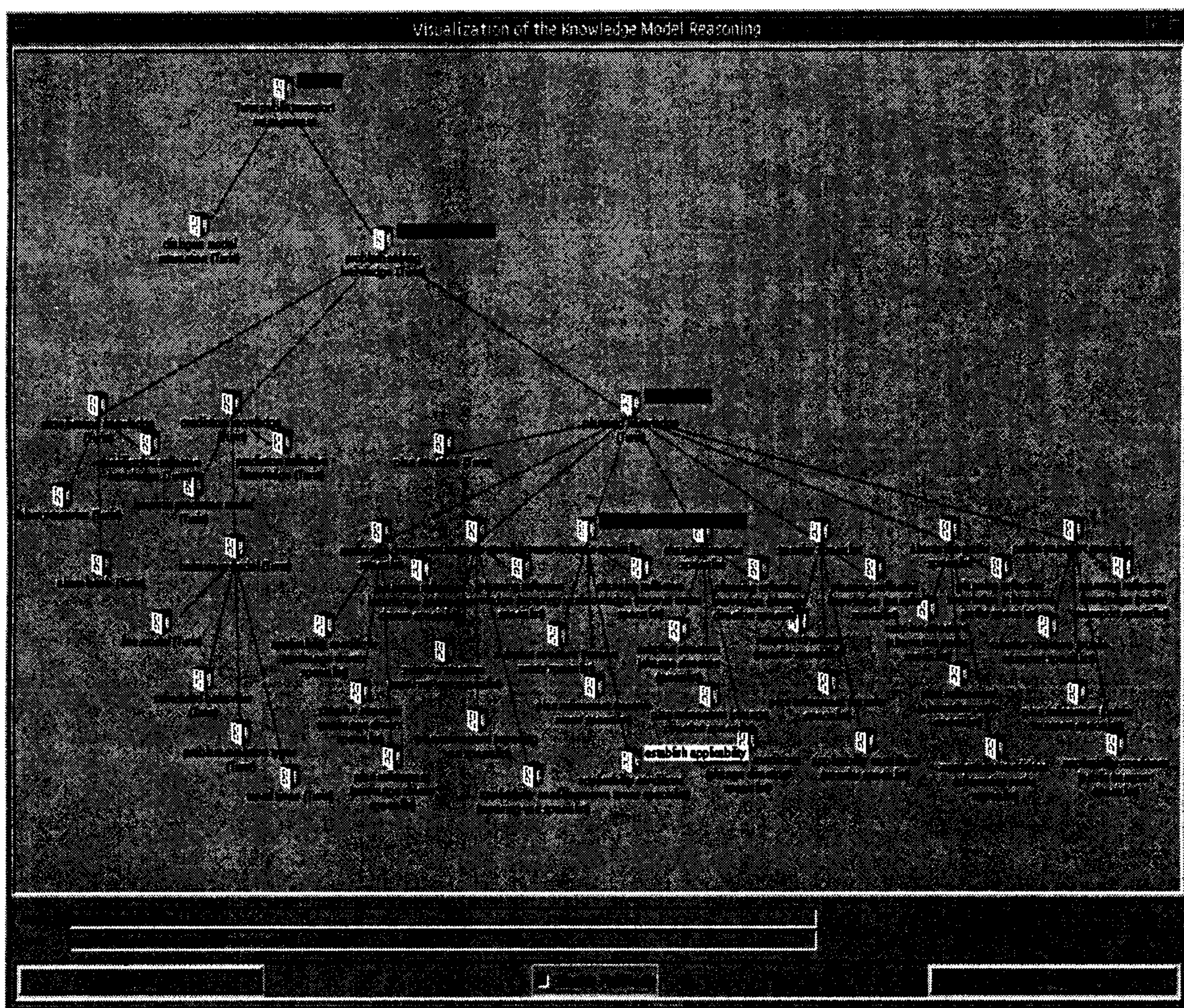


Figura 8.15: Monitorización de la ejecución del modelo de gestión de transporte público

## 8.5 Conclusiones

El sistema mostrado en este capítulo es el primer prototipo del proyecto FLUIDS y aunque constituye una implementación parcial del modelo de



gestión de transporte público descrito en el capítulo 7, se considera suficiente para mostrar las posibilidades y viabilidad computacional de la arquitectura ROHCI. En la actualidad este prototipo está en fase de ampliación e instalación *on-line* en el centro de control de transporte público de Turín.

# CONCLUSIONES





## 9. Evaluación

---

En los capítulos anteriores se ha presentado la arquitectura ROHCI (*Reasoning Oriented to Human-Computer Interaction*) orientada a mejorar el nivel de interacción entre usuarios y sistemas, y se ha visto cómo la arquitectura propuesta se ha aplicado sobre un problema real como es la gestión del transporte público en la ciudad de Turín. Este capítulo se centra en evaluar en qué medida la solución aportada por esta arquitectura satisface los principios para una interacción usuario-sistema avanzada, definidos en el capítulo 5, desde las dos perspectivas desarrolladas en el estado del arte:

- por una parte, se comparan las posibilidades comunicativas que ofrecen las propuestas más relevantes presentadas en el estado del arte sobre modelos dinámicos de conocimiento y las que proporciona ROHCI, y en particular su aplicación en el sistema FLUIDS.
- por otra parte, se analiza en qué medida los generadores de presentaciones inteligentes pueden verse favorecidos por el soporte de un modelo de resolución de problemas como el propuesto en ROHCI.

### 9.1 Evaluación desde la Perspectiva de los Sistemas Inteligentes

Esta sección compara TIPS, MODEL-K y TroTelC con la arquitectura ROHCI, y su aplicación en el sistema FLUIDS, con respecto a las posibilidades que ofrece cada una de ellas de satisfacer los principios de interacción usuario-sistema. La evaluación se centra en estas soluciones de modelización dinámica de conocimiento por ser éstas las que pueden proporcionar un nivel de interacción más acorde con los objetivos que se plantean en esta tesis. Otras arquitecturas más simples ofrecen posibilidades de interacción más limitadas por lo que no se consideran en este análisis.



## ♦ CONSCIENCIA

Se entiende como la capacidad que tiene el sistema para asumir la responsabilidad sobre sus conclusiones, estableciendo claramente el alcance y soporte de las mismas. Esta capacidad reside en el grado de clasificación y estructuración del conocimiento del dominio problema que se haya incluido en el sistema. A mayor grado de organización mayor nivel de conocimiento del sistema sobre sus propias posibilidades, y por tanto mayor capacidad de autocrítica.

A la hora de evaluar este aspecto, se distinguen tres niveles de organización del conocimiento:

- el sistema está concebido como un modelo de una parcela de la realidad en el que se distingue claramente el conocimiento del dominio del modo de razonar con él para resolver problemas. Así se da la posibilidad de que el sistema acompañe sus conclusiones con las piezas de conocimiento del dominio en las que ha apoyado su razonamiento. Este nivel de organización se correspondería con el que ofrecen los sistemas basados en el conocimiento convencionales.
- el modelo de resolución de problemas incorpora una organización explícita del conocimiento del dominio. Así se ponen de manifiesto los diferentes tipos de conocimiento presentes en el dominio problema y las estrategias de razonamiento que pueden emplearse con este conocimiento para resolver problemas en ese dominio. De esta manera, el usuario puede recibir como respuesta del sistema no sólo las conclusiones acompañadas de las piezas de conocimiento del dominio utilizadas, sino también el rol que cada una de estas piezas ha jugado en el razonamiento. Este nivel de organización es el que presentan los sistemas basados en el conocimiento de segunda generación.
- el sistema puede razonar sobre sus propias capacidades y por tanto distingue entre conocimiento de resolución de problemas y conocimiento reflexivo. De esta manera, el tipo de respuesta que podría ofrecer el sistema a partir del nivel anterior, se vería reforzada con los argumentos que han apoyado el uso de una determinada estrategia de razonamiento.

En el análisis de este tercer nivel de especificación del conocimiento cabe considerar también la presencia de estructura en su configuración. Así se dota al sistema de la capacidad de justificar sus decisiones a metanivel, es decir, podría explicar los pasos dados para llegar a tomar esa decisión, cuáles eran sus alternativas, qué criterios se aplicaron para elegir una de ellas o por qué las demás no fueron elegidas.

De acuerdo con los niveles anteriores, la valoración de los sistemas TIPS, MODEL-K, TroTelC y ROHCI quedaría como sigue:

- ⇒ Todas las soluciones se basan en modelos estructurados de conocimiento del dominio problema, por lo que ya incorporan el primer nivel básico de organización. TIPS está organizado como una estructura de tareas genéricas, MODEL-K se basa en la metodología KADS para construir sistemas basados en el conocimiento, TroTelC aplica el enfoque de Componentes de la Experiencia y ROHCI conceptualmente es similar a TIPS aunque en su implementación en el sistema FLUIDS aplica el modelo de unidades cognitivas de KSM.
- ⇒ La presencia de un nivel de razonamiento reflexivo se da en los cuatro sistemas analizados. TIPS, TroTelC y ROHCI pueden decidir dinámicamente el método que deben aplicar en cada paso de inferencia. MODEL-K tiene esta capacidad limitada al poder hacer razonamiento metanivel sólo sobre las tareas de más alto nivel del modelo de objetos, aunque este aspecto no le impide ofrecer justificaciones adecuadas de las acciones que realiza a este nivel.
- ⇒ El conocimiento metanivel está estructurado en todos los casos salvo en TroTelC. TIPS lo organiza en torno al mecanismo espónsor-selector, MODEL-K aplica a metanivel la misma descomposición en niveles de KADS que utiliza a nivel objeto, y ROHCI lo tiene organizado sobre una jerarquía de especialistas. En el caso de TroTelC, el conocimiento reflexivo no tiene estructura propia y se encuentra integrado en el conocimiento a nivel objeto.

De estas consideraciones se concluye que el nivel de consciencia que pueden alcanzar ROHCI, TIPS y MODEL-K es similar, siendo superior en los tres casos al de TroTelC.



## ♦ TRANSPARENCIA

La transparencia de un sistema se evalúa por las facilidades que ofrece para acceder a los elementos en los que reside el conocimiento del sistema. Al poner a la vista del usuario el modelo de comprensión del dominio que tiene el sistema, se hace posible que éste posea una visión integral de la aplicación que facilita la asimilación de las conclusiones del sistema, genera confianza en sus respuestas y le da la capacidad de adivinar, o al menos no sorprenderse, del tipo de funcionalidades que puede ofrecerle el sistema. Además, así se da al usuario la posibilidad de relacionar con facilidad los resultados obtenidos recientemente por el sistema con los que previsiblemente éste le ofrezca a continuación, contribuyendo de esta manera a que el sistema se constituya como un apoyo real del usuario en la medida en la que le proporciona información de detalle que espera para validar sus propias intuiciones.

A la hora de evaluar la accesibilidad de un sistema pueden considerarse dos niveles, ambos aplicables a los dos niveles fundamentales de especificación de conocimiento, metanivel y objeto, considerados:

- se puede acceder a una representación explícita de las estrategias de razonamiento aplicadas por el sistema;
- se puede acceder al conocimiento del dominio, es decir, al contenido de las bases de conocimiento del sistema.

Con respecto a las cuatro soluciones objeto de comparación, TIPS, MODEL-K, TroTelC y ROHCI, la evaluación de este aspecto daría como resultado lo siguiente:

- ⇒ A nivel objeto, las cuatro propuestas dan acceso a su modelo de razonamiento al proporcionar una visión de sus respectivas jerarquías de tareas, y también a sus diferentes bases de conocimiento.
- ⇒ A metanivel pueden observarse algunas diferencias. Con respecto a la accesibilidad de la estrategia de inferencia empleada, tanto TIPS como MODEL-K y ROHCI proporcionan una representación explícita de la línea de razonamiento que aplican para tomar decisiones a este nivel,

mientras que en TroTelC este conocimiento está integrado en la jerarquía de tareas del nivel objeto.

Por otra parte, algo similar pasa con el conocimiento empleado para hacer este razonamiento reflexivo, ya que nuevamente TIPS, MODEL-K y ROHCI emplean formulaciones explícitas del conocimiento metanivel mientras que TroTelC oculta parte de este conocimiento. En concreto, en TroTelC no puede accederse al conocimiento aplicado para seleccionar qué método ha de usarse de entre un conjunto de métodos aplicables.

Entonces, se entiende que el grado de transparencia de ROHCI, TIPS y MODEL-K es similar y satisfactorio, pero el de TroTelC es inferior al ocultar tanto su estrategia como el conocimiento que emplea en el razonamiento metanivel.

#### ♦ FAMILIARIDAD

Este aspecto se entiende como la capacidad del sistema para expresar lo que sabe en forma comprensible al usuario. En este sentido, el objetivo es analizar la expresividad del conocimiento de que dispone el sistema, y no la de sus conclusiones o respuestas, puesto que a los sistemas inteligentes se les asume la capacidad de formular sus conclusiones en forma suficientemente comprensible para sus usuarios. Por tanto, ya que se están considerando sistemas abiertos, la familiaridad se refiere a la formulación del conocimiento empleado para generar esas conclusiones y no a las conclusiones en sí.

La familiaridad es un elemento de soporte fundamental de los principios de consciencia y transparencia, puesto que:

- por una parte, la posibilidad de mostrar las diferentes piezas de conocimiento que han intervenido en la obtención de las respuestas del sistema se debe completar con una expresión de las mismas en el lenguaje empleado habitualmente por los conocedores del dominio problema;
- por otra parte, la accesibilidad a la estructura de conocimiento del sistema no resulta verdaderamente útil - desde la perspectiva del usuario



- si la expresión de esa estructura y su contenido no puede ser comprendida por el usuario. Si la formulación de éste conocimiento se hace empleando un lenguaje cercano a los usuarios se da la posibilidad real de acceder directamente al conocimiento de la aplicación y modificarlo de acuerdo con la experiencia adquirida.

La evaluación de este aspecto se hace en función de la cercanía que haya podido alcanzarse en el lenguaje de especificación del conocimiento del modelo a modos de expresión más próximos al usuario. Para ello puede tenerse en cuenta tanto la capacidad expresiva del lenguaje empleado como la terminología utilizada en sus expresiones. Con respecto al primer aspecto, puede distinguirse entre lenguajes de especificación de conocimiento de dominio de tipo atributo-valor y de tipo relacional, puesto que estos últimos hacen posible simplificar la declaración de ese conocimiento a la vez que ponen de manifiesto explícitamente las relaciones entre los conceptos del dominio. Además, con respecto al lenguaje de formulación del modelo de inferencia, también pueden distinguirse entre los de tipo procedural frente a los de tipo declarativo, siendo estos últimos expresivamente más potentes.

Al aplicar estos criterios sobre las soluciones a comparar se obtienen las siguientes conclusiones:

- ⇒ con respecto al lenguaje de formulación del conocimiento del dominio, de TIPS y TroTelC no se tiene información suficiente para calificar el tipo de lenguaje empleado. MODEL-K es un caso similar ya que no es un sistema implementado, sino una solución de análisis de sistemas con capacidades reflexivas, por lo que tampoco cabe considerar este aspecto. En el caso de ROHCI, su implementación en el sistema FLUIDS emplea el lenguaje de especificación de conceptos Concel, proporcionado por KSM, que es un lenguaje relacional. Estos conceptos se emplean en la especificación de otras estructuras de conocimiento como reglas, marcos, restricciones o tablas.
- ⇒ con respecto al lenguaje de formulación del modelo de inferencia, tanto MODEL-K como TroTelC utilizan un lenguaje procedural mientras que TIPS aplica un lenguaje declarativo. En este caso, el lenguaje empleado para definir el modelo de conocimiento de FLUIDS, también proporcionado por el KSM, es un lenguaje declarativo, llamado Link,

que expresa el conocimiento de control mediante reglas de producción. Estas reglas tienen en su parte izquierda referencias a valores de estados de control obtenidos durante la ejecución y en la parte derecha definen la secuencia de tareas u objetivos que deben realizarse a continuación.

Entonces, se entiende que la familiaridad que alcanza ROHCI, en su implementación en el sistema FLUIDS, es superior a la de las otras propuestas porque se conocen los lenguajes empleados en la especificación del conocimiento del sistema, Concel y Link, y por la expresividad de los mismos. En el caso de TIPS, se considera un nivel de familiaridad superior a los de MODEL-K y TroTelC ya que, aunque en los tres casos se desconoce el lenguaje de formalización del conocimiento del dominio, TIPS utiliza un lenguaje declarativo para especificar el modelo de inferencia mientras que MODEL-K y TroTelC utilizan un lenguaje procedural.

#### ◆ FLEXIBILIDAD

Este aspecto se entiende como la capacidad del sistema para mantener diálogos no lineales con sus usuarios, de forma similar a los que se producen entre personas, en los que en ocasiones los interlocutores rectifican hipótesis anteriores o simplemente reconsideran su posición. La gestión eficiente de este tipo de diálogos pasa por la disponibilidad de una memoria que contenga una representación del modelo de conocimiento común a usuario y sistema, obtenido con el desarrollo del diálogo, pero sobre todo por dotar a esa memoria de un mecanismo gestor que sepa cómo mantener consistente ese estado de información, tal como se produce en ROHCI.

Atendiendo a esta definición, TIPS, TroTelC y MODEL-K podrían llegar a proporcionar el nivel de flexibilidad que les permite su capacidad para reflexionar sobre alternativas de solución a un mismo problema, pero no el tipo de flexibilidad con el que cabría reconducir la dirección del diálogo. En el caso de ROHCI, esta forma de entender la flexibilidad de la interacción usuario-sistema está soportada por la presencia de una memoria del diálogo gestionada de forma no monótona por el mecanismo ATMS, que ofrece por una parte la posibilidad de identificar y resolver inconsistencias en el estado de creencias comunes a usuario y sistema generado con la interacción, y por otra, da soporte al planteamiento de un amplio abanico de preguntas: consultas sobre



conclusiones extraíbles a partir de los datos recientes enviados por el sistema de información, consultas sobre resultados previsibles ante la presencia de diferentes hipótesis que pueden plantearse en forma conjuntiva o disyuntiva, o solicitudes de revisión de conclusiones previas a la luz de contradicciones en planteamientos hipotéticos considerados con anterioridad.

Entonces, se considera que el nivel de flexibilidad que puede alcanzar ROHCI es superior al de las otras tres soluciones fundamentalmente porque a la hora de decidir la forma de resolver un problema o realizar una tarea, el sistema no analiza el problema de forma aislada sino que considera a éste situado dentro del contexto de un diálogo usuario-sistema que cambia y evoluciona.

#### ◆ VERSATILIDAD

Se entiende como la capacidad del sistema para adaptarse a las características del contexto del diálogo que mantenga con un usuario. Este contexto debe analizarse desde dos perspectivas: (i) la que considera los aspectos del problema que hay que resolver y el contexto de información en el que ha de resolverse, y (ii) la que contempla las características de los usuarios con los que el sistema puede interactuar. La capacidad de adaptación del sistema depende de sus posibilidades para reflexionar sobre ambos aspectos: el problema y el usuario; y su objetivo será identificar la forma más adecuada de obtener la solución al problema - la respuesta a una pregunta - en función de las necesidades del usuario.

Entonces, al estudiar estos aspectos sobre las soluciones que se están evaluando se obtienen las siguientes conclusiones:

- Con respecto a la capacidad de razonamiento a metanivel mediante el que se decide dinámicamente cuáles son los métodos de resolución de problemas que deben aplicarse para generar una respuesta: MODEL-K tiene capacidad de razonamiento reflexivo, pero esta capacidad no está orientada a seleccionar dinámicamente entre distintos métodos de resolución de problemas sino a estudiar características del problema a resolver antes de ponerse a resolverlo, dando la oportunidad de replantear aspectos del problema para que pueda ser resuelto por el modelo de razonamiento. ROHCI, TIPS y TroTelC tienen la posibilidad

de elegir dinámicamente el método a aplicar, pero pueden establecerse distinciones entre ellos atendiendo a:

- ⇒ el nivel, sobre la estructura de tareas del modelo, en el que pueden tomarse decisiones metanivel, ya que en el caso de MODEL-K sólo se toman decisiones sobre las tareas de más alto nivel, sin intervenir en aquellas otras que forman parte de las etapas intermedias del razonamiento, mientras que TIPS, TroTelC y ROHCI pueden seleccionar el método a aplicar en cualquier punto de la estructura de inferencia en el que se tengan alternativas de solución.
- ⇒ los criterios empleados para evaluar la adecuación de los distintos métodos de resolución de problemas disponibles. La valoración de la adecuación de un método se puede hacer de forma local al método, asignándole un valor en función del contexto de la operación, o bien se puede estimar comparando cada método con los demás con respecto a aspectos relevantes del contexto de la operación.

En este sentido, el sistema TIPS aplica criterios de valoración locales al método y cuantitativos que posteriormente son utilizados por el selector de métodos para elegir el más adecuado. Esta aproximación tiene el problema de que cualquier extensión o modificación del conjunto de métodos disponibles es costosa porque requiere revisar las valoraciones asignadas a cada uno de los métodos que compiten con el nuevo. Por otra parte, ROHCI valora la adecuación de un método comparándolo con el resto de métodos disponibles en función de su capacidad para satisfacer los requisitos del contexto de operación. Así, la incorporación de un nuevo método para resolver una clase de tarea sólo supone situarlo en la posición adecuada con respecto a los demás. En lo que respecta a TroTelC, tal como se ha mencionado antes, este sistema mantiene oculta su estrategia y conocimiento para seleccionar entre métodos aplicables por lo que no es posible evaluar este aspecto.

- Con respecto a la presencia y aplicación de conocimiento sobre características de los usuarios que refuerce las decisiones sobre el modo de proporcionar una respuesta, ninguna de las soluciones del estado del



arte objeto de esta comparación considera ningún aspecto del usuario a la hora de decidir cómo resolver un problema. Por contra, ROHCI condiciona su forma de encontrar las respuestas que solicita el usuario al conocimiento de que dispone sobre él y sus intenciones comunicativas. En concreto, el particular modelo de conocimiento sobre los usuarios que utiliza ROHCI incluye información sobre: (i) tipos de usuarios y objetivos comunicativos, representadas en los marcos de escenarios de interacción y, (ii) estado de conocimiento, reflejado en la memoria de la interacción, al entender que el usuario sabe lo que el sistema le ha ido transmitiendo a lo largo del diálogo.

Al entender que la versatilidad está relacionada con la capacidad del sistema para dar a cada usuario la información que requiere en función de sus características, pasa a ser relevante también considerar la información que el sistema 'no' debe proporcionar a ese usuario por esas mismas características. Esto significa que el sistema ha de ser capaz de adaptar su contribución y su papel en el diálogo con el usuario a la autoridad de su interlocutor. Por ejemplo, el sistema de gestión de transporte público FLUIDS, cuyo modelo conceptual es el de ROHCI, incluye la posibilidad de obtener respuestas a diferentes niveles de abstracción tanto para caracterizar el estado de una línea como para presentar las propuestas de acciones de control, que se seleccionan dependiendo de las preferencias y características de los usuarios, como el nivel de experiencia o el objetivo comunicativo.

Entonces, del análisis de la versatilidad de un sistema desde su capacidad para reflexionar sobre el modo de resolver un problema sin considerar aspectos del usuario, se entiende que el nivel de versatilidad de ROHCI sería similar al de TIPS, siendo el de ambos superior al de MODEL-K y TroTelC. Ahora bien, si añadimos además la posibilidad de utilizar en esa reflexión algún conocimiento sobre los usuarios entonces las posibilidades que ofrece ROHCI son superiores a las de los anteriores.

#### ◆ AGILIDAD

Se entiende como la capacidad del sistema para entablar y mantener con sus usuarios conversaciones desarrolladas de forma eficiente. Este aspecto de eficiencia reside en la disponibilidad de una memoria de la interacción en la que resida un conjunto consistente de las conclusiones alcanzadas por el

sistema a lo largo de su diálogo con el usuario. Así, esta memoria puede facilitar la operación del resolvidor de problemas desde dos perspectivas:

- por una parte, funciona como un depósito de información que este resolvidor puede consultar para extraer información generada con anterioridad, evitando de esta manera volver a generarla y,
- por otra parte, puede prevenir al resolvidor respecto a la presencia de inconsistencias entre creencias ya establecidas y contenidas en la memoria e hipótesis consideradas en el planteamiento del problema a resolver, evitando de esta manera que éste lleve a cabo inferencias que resultarían infructuosas.

En este sentido, las propuestas analizadas en el estado del arte no disponen de nada más que una memoria de trabajo pero no una memoria de diálogo como la descrita, básicamente porque no entienden la interacción con sus usuarios como un proceso dirigido por un diálogo. Este aspecto es considerado explícitamente en ROHCI, cuyo modelo concibe una memoria de la interacción usuario-sistema con las características anteriores. Por otra parte, el modelo de memoria que proporciona un ATMS, como el que se utiliza en ROHCI, tiene una importante ventaja adicional como es la posibilidad de obtener rápidamente información sobre lo que es deducible del estado de creencias del sistema, al tener el conocimiento representado en un retículo de nodos ligados por relaciones causales, que pueden ser tanto hechos contrastados como contradicciones entre hipótesis. Este aspecto contribuye a la eficiencia del sistema al dar la posibilidad de tomar acciones correctivas a priori en el caso en el que se planteen hipótesis inconsistentes con otras declaradas previamente, antes de ejecutar los procesos de razonamiento que llevarían a extraer las mismas conclusiones.

La tabla presentada en la figura 9.1 sirve de resumen al análisis comparativo realizado desde la perspectiva de los Sistemas Inteligentes.



Conceptos evaluados y aspectos analizados	TIPS	MODEL-K	TROTELC	ROHCI	Observaciones del análisis comparado
<b>CONSCIENCIA</b> Niveles de estructuración del conocimiento: <ul style="list-style-type: none"><li>• Distinción entre las piezas de conocimiento del dominio y el modo de usarlas</li><li>• Definición explícita de un modelo de organización del conocimiento del dominio</li><li>• Definición explícita de un modelo de conocimiento reflexivo</li></ul>	+	+	0	+	<ul style="list-style-type: none"><li>• Nivel máximo de estructuración del conocimiento en TIPS, MODEL-K y ROHCI</li><li>• Conocimiento reflexivo de TroTelC no está estructurado</li></ul>
<b>TRANSPARENCIA</b> Niveles de acceso al conocimiento del sistema: <ul style="list-style-type: none"><li>• Acceso a una formulación explícita de las estrategias de razonamiento a nivel de objetos y a metanivel</li><li>• Acceso al conocimiento del dominio a nivel de objetos y a metanivel</li></ul>	+	+	0	+	<ul style="list-style-type: none"><li>• TIPS, MODEL-K y ROHCI dan acceso a las estrategias de razonamiento y al conocimiento del dominio tanto a metanivel como a nivel de objetos</li><li>• TroTelC oculta ambos tipos de conocimiento a metanivel</li></ul>
<b>FAMILIARIDAD</b> Expresividad del lenguaje utilizado para: <ul style="list-style-type: none"><li>• Formular el conocimiento del dominio: atributo-valor vs. relacional</li><li>• Formular el modelo de inferencia: procedural vs. declarativo</li></ul>	0	-	-	+	<ul style="list-style-type: none"><li>• Lenguaje de formulación de conocimiento del dominio de ROHCI es relacional. De TIPS, MODEL-K y TroTelC no se dispone de información suficiente.</li><li>• Lenguaje de formulación del modelo de inferencia de MODEL-K y TroTelC es procedural. El de TIPS y ROHCI es declarativo.</li></ul>
<b>FLEXIBILIDAD</b> Capacidad para desarrollar diálogos no lineales: <ul style="list-style-type: none"><li>• Bajo planteamientos alternativos</li><li>• Revisando hipótesis aceptadas previamente</li></ul>	-	-	-	+	<ul style="list-style-type: none"><li>• Las arquitecturas de TIPS, MODEL-K y TroTelC podrían llegar soportar el razonamiento bajo planteamientos alternativos, pero no la revisión de hipótesis.</li><li>• ROHCI soporta ambas cosas con la ayuda de la memoria de la interacción gestionada por el ATMS.</li></ul>
<b>VERSATILIDAD</b> Capacidad de adaptación a las características de un diálogo: <ul style="list-style-type: none"><li>• Atendiendo al contexto del problema</li><li>• Atendiendo a características de los usuarios</li></ul>	0	-	0	+	<ul style="list-style-type: none"><li>• MODEL-K no tiene capacidades específicas de selección dinámica de métodos. TIPS, TroTelC y ROHCI pueden seleccionar entre métodos alternativos en cualquier etapa del razonamiento. TroTelC oculta parte de este conocimiento.</li><li>• ROHCI tiene conocimiento de los tipos de usuarios, su estado de conocimiento y sus objetivos comunicativos. TIPS, MODEL-K y TroTelC no disponen de ningún conocimiento sobre los usuarios.</li></ul>
<b>AGILIDAD</b> Capacidad para desarrollar diálogos eficientes: <ul style="list-style-type: none"><li>• Acceso a información ya generada en el diálogo</li><li>• Detección de información inconsistente con el estado de información.</li></ul>	-	-	-	+	<ul style="list-style-type: none"><li>• ROHCI dispone de una memoria de la interacción con un conjunto consistente de creencias alcanzadas en el diálogo. El mecanismo ATMS hace posible prevenir y resolver inconsistencias en el conjunto de creencias.</li><li>• TIPS, MODEL-K y TroTelC no tienen memoria.</li></ul>

Figura 9.1: Tabla comparativa desde la perspectiva de los Sistemas Inteligentes



## 9.2 Evaluación desde la Perspectiva de los Interfaces de Usuario Inteligentes

En esta sección se analizan los beneficios que la arquitectura ROHCI puede ofrecer a los interfaces de usuario inteligentes, de cara a la generación de mejores presentaciones, en un contexto de operación en el que el objetivo es ofrecer un buen nivel de interacción usuario-sistema. Dichos beneficios residen fundamentalmente en los siguientes aspectos:

♦ *Mayor riqueza de la información a presentar*

El generador de presentaciones puede disponer de información clasificada de acuerdo con las necesidades de comunicación. La naturaleza de la información que puede proporcionarle ROHCI es de dos tipos:

- (i) Información relativa a los procesos de resolución de problemas: no sólo se dispone de los datos utilizados y obtenidos durante un proceso de razonamiento, sino que éstos se acompañan del rol que ha desempeñado cada uno de estos datos en el razonamiento, y la línea argumental seguida para alcanzar esas conclusiones, tanto en lo que respecta a la estrategia de razonamiento aplicada como a la elección de dicha estrategia, lo cual permite aprovechar dicha información adicional para mejorar la comunicación.
- (ii) Información sobre criterios de interacción relativos al modo en el que el sistema debe orientar la generación de la información en función de las características o necesidades comunicativas de los usuarios. Esta información se genera gracias a la posibilidad que ofrece la arquitectura ROHCI par razonar sobre formas de interacción con el modelo de Escenarios de Interacción. La disponibilidad de dicha información permite una mejor elección de las formas de presentación dado que se refieren a características que expresan el modo en el que debe realizarse la comunicación.

Dado que las últimas generaciones de presentadores inteligentes disponen de modelos declarativos de configuración de presentaciones en los que se definen tanto estrategias de elaboración de presentaciones como aspectos relativos a modelos de usuarios, el tener acceso a esta información que



proporciona el modelo de resolución de problemas puede enriquecer la definición de dichos modelos de generación de presentaciones, redundando al final en presentaciones de mayor calidad.

♦ *Mayor facilidad en la interpretación de las demandas del usuario*

La arquitectura ROHCI simplifica el proceso de interpretación de las demandas comunicativas de un usuario al incorporar un análisis sobre las formas en las que el usuario puede expresar sus necesidades comunicativas, es decir, al incluir dentro del modelo de resolución de problemas la semántica del diálogo con los usuarios.

ROHCI utiliza un modelo explícito de la interacción usuario-sistema cuyo objetivo es comprender qué es lo que desea el usuario y cómo necesita la información que demanda en función de sus características, lo cual descarga al interfaz de usuario de procesos que hasta ahora habrían tenido que realizar si hubiese deseado dar el mismo servicio, recorriendo por sí mismo el camino que va desde la interpretación de las necesidades del usuario (también en los aspectos de interacción) hasta la identificación de los procesos que deberían aplicarse para generar la información solicitada.

♦ *Mayor facilidad en la elaboración de respuestas satisfactorias*

De forma análoga a como sucede en la interpretación de las demandas de información de un usuario, el modelo de resolución de problemas de ROHCI genera sus respuestas con una determinada intención comunicativa cercana a las necesidades del diálogo, lo cual nuevamente simplifica o ayuda al interfaz de usuario en la tarea de seleccionar y configurar la información que ha de presentarse a un usuario en función de sus necesidades.

♦ *Reducción del esfuerzo en el tratamiento del contexto del diálogo*

Desde la perspectiva de una buena interacción usuario-sistema, esta interacción necesita de una memoria del diálogo que agilice la comunicación y en la que se consideren tantos los aspectos tradicionalmente tratados por los interfaces, como son los relativos a la coherencia de los elementos mostrados en el interfaz a lo largo del tiempo, como de la información que ya se ha ofrecido a los usuarios. ROHCI da soporte a esta memoria con un

mecanismo que no es sólo una memoria en donde reside la información generada a lo largo de un diálogo en su contexto, sino que además esta memoria tiene dos características especialmente significativas:

- (i) ofrece al usuario la posibilidad de formular sus preguntas de forma incremental, es decir, sin necesidad de especificar el conjunto completo de hipótesis que considera en su consulta, ya que el propio sistema tiene la capacidad de acumular o discriminar entre las hipótesis planteadas por el usuario durante el transcurso del diálogo, y
- (ii) tiene asociado un procedimiento de revisión que le permite la retracción de creencias obtenidas durante el diálogo manteniendo el conjunto de creencias consistente y apoyando de esta manera también la consistencia de la información mostrada por el interfaz.

En resumen, en términos generales podría decirse que el soporte de una arquitectura como ROHCI facilita el que los interfaces de usuario puedan centrar su atención más en los aspectos de presentación de información y menos en los aspectos relativos a la comunicación, puesto que éstos ya son tratados en el modelo de resolución de problemas.





# 10. Conclusiones

---

En este capítulo se presenta un resumen de las conclusiones alcanzadas en el desarrollo de este trabajo de investigación.

## 10.1 Contribuciones

El objetivo fundamental de esta tesis ha sido el estudio y propuesta de formas más avanzadas de interacción entre usuarios y sistemas, en el ámbito de entornos de operación de toma de decisiones.

Las principales aportaciones de esta tesis en este sentido son las siguientes:

- (1) Análisis de las soluciones de interacción usuario-sistema actuales desde dos perspectivas: (i) la que plantean los sistemas inteligentes basados en modelos estructurados de conocimiento que persiguen acercarse a los usuarios mediante la emulación de su modelo de comprensión del dominio problema y, (ii) la que ofrecen los interfaces de usuario inteligentes orientados a potenciar la capacidad expresiva de los sistemas mediante el uso de múltiples medios.

En este estudio se llega a la conclusión de que, en ambas disciplinas, existe un desequilibrio muy significativo entre el esfuerzo y la complejidad dedicada al modelo propio y el que corresponde al modelo complementario, considerando que obviamente interfaces y aplicaciones forman una unidad. Así, los sistemas inteligentes están especializados en la resolución de problemas pero no necesariamente en dirigir esa capacidad hacia los intereses comunicativos de un usuario y, los interfaces de usuario inteligentes saben cómo comunicar información a un usuario pero sin profundizar en la naturaleza de la información que manejan y en la forma en la que ésta se genera.



- (2) A la vista de las deficiencias observadas, se plantea la búsqueda de una solución integradora que dé continuidad a los modelos de resolución de problemas y los de presentación, de forma que las presentaciones puedan diseñarse a la medida del conocimiento del sistema y el razonamiento del modelo de resolución de problemas tenga en cuenta las características de los usuarios a la hora de generar la información que requieren.

Con este objetivo, se presenta un nuevo concepto de Interfaz Inteligente aglutinante de ideas provenientes de los sistemas inteligentes y de los interfaces de usuario inteligentes, concebido bajo la máxima: *la forma de conversar condiciona la forma de resolver problemas*, y que se fundamenta en un conjunto de principios de la interacción usuario-sistema: consciencia, transparencia, familiaridad, versatilidad, flexibilidad y agilidad. Estos principios definen las características deseables en el modelo de interacción entre un sistema y los usuarios articulado en torno a un diálogo o secuencia de preguntas y respuestas mediante las que el usuario coopera con el sistema para alcanzar sus objetivos.

- (3) Se propone una arquitectura, llamada ROHCI, mediante la que pueden definirse sistemas que soporten el modelo de interacción usuario-sistema definido previamente. Los elementos fundamentales de esta arquitectura son los siguientes:
  - Un *Modelo de Interacción* estructurado por tipos de preguntas que define los elementos en torno a los que se desarrollará el proceso comunicativo.
  - Un *Espacio de Resolvedores de Problemas* concebido como un modelo de conocimiento estructurado jerárquicamente donde reside el conocimiento de resolución de los problemas propios del dominio de la aplicación y las piezas software capaces de llevar a cabo el razonamiento.

Para desarrollar este elemento se ha hecho uso de las técnicas más avanzadas del área de Ingeniería del Conocimiento sobre modularización y estructuración de conocimiento, que

proporcionan una solución para organizar en módulos especializados (tareas, métodos, áreas de conocimiento) los distintos componentes que intervienen en la resolución de problemas.

Dicha organización ofrece además los medios de expresión adecuados para disponer de una representación explícita del propio modelo de conocimiento, lo que sirve de base para especificar el comportamiento reflexivo del sistema.

- Un *Modelo de Razonamiento Metanivel*, donde reside la capacidad reflexiva del sistema, que navega en el espacio de resolución de problemas para proponer, en función de las características del usuario y del contexto del diálogo, el tipo de respuestas que deben proporcionarse al usuario y el modo de obtenerlas. Este módulo reflexivo se apoya en dos tipos de conocimiento:
  - (i) Conocimiento sobre escenarios de interacción con los diferentes tipos de usuarios en los que se representan los criterios que deben guiar la búsqueda de la respuesta esperada por el usuario en función de las etapas previas del diálogo y de las intenciones comunicativas y características del usuario. Estos criterios se definen como atributos de interacción y describen características de los métodos de resolución de problemas deseables desde la perspectiva de la interacción.

La definición de estos escenarios de interacción se ha inspirado en las técnicas de modelización de usuarios basadas en el uso de estereotipos, aunque su aplicación no se ha orientado hacia la modelización del conocimiento de tipos de usuarios, sino más bien a modelizar aspectos de la interacción que debe considerar el sistema a la hora de decidir cómo satisfacer las demandas de los usuarios que están en consonancia con las características y necesidades comunicativas de cada momento.



- (ii) Conocimiento especializado en el diseño dinámico de líneas de razonamiento que resuelvan los problemas planteados por el usuario durante el diálogo. Este diseño se obtiene como agregación de resolvedores parciales de problemas elegidos en base a los atributos de interacción decididos anteriormente y el estado de la interacción.

La caracterización de este conocimiento se ha apoyado en métodos para diseño automático orientado hacia la selección dinámica de métodos de resolución de problemas. En particular, se ha utilizado una solución de diseño que concibe el modelo de conocimiento de diseño como una colección estructurada de especialistas que se han formulado de manera que seleccionen el método más adecuado en función de las necesidades de la interacción.

Este módulo de razonamiento reflexivo constituye el núcleo de la arquitectura ROHCI, siendo responsable de adoptar las decisiones relativas al modo de interaccionar con los usuarios, desde la interpretación de las necesidades comunicativas de esos usuarios hasta la determinación de la forma en que han de resolverse los problemas planteados por ellos para satisfacer esas necesidades.

- Un *Contexto de la Interacción*, donde reside el conjunto consistente de creencias comunes al sistema y el usuario generado durante la interacción, que está gestionado por un sistema de mantenimiento de la verdad de tipo ATMS, y con el que se asegura la coherencia de la interacción usuario-sistema en el tiempo.

Esta arquitectura aporta la suficiente flexibilidad para generar de forma automática la forma de responder a las preguntas de un usuario, utilizando una estructura de representación análoga al modo de organizar el conocimiento en el Espacio de Resolvedores de Problemas. Además, por sus características, puede integrarse con cualquier sistema de ayuda a la decisión en tiempo real, ofreciéndole soporte en su interacción con los usuarios, tras extraer un modelo de la organización y capacidades de resolución de problemas del sistema en

términos similares a los que ofrece el Espacio de Resolvedores de Problemas. El objeto de este modelo es proporcionar una visión abstracta de las capacidades del sistema de cara a la especificación del Modelo de Razonamiento Metanivel y del Modelo de Interacción.

- (4) Para mostrar la viabilidad de la arquitectura se presenta el diseño de un sistema de ayuda a la decisión en el dominio de gestión de transporte público. Se trata de un ejemplo de dimensión real tomado de un proyecto en curso ([FLUIDS, 96]) cuyo objetivo es construir un interfaz inteligente que de soporte a los operadores del centro de control de transporte público (autobuses y tranvías) de la ciudad de Turín. El desarrollo de dicho sistema se ha facilitado con el uso del entorno software KSM ([Molina, Cuenca, 95], [Cuenca, Molina, 94, 96, 97]) que proporciona los medios necesarios para formular y hacer operativos modelos estructurados de conocimiento.

Las ventajas que aporta una arquitectura como ROHCI, basada en una organización estructurada y flexible de conocimiento diseñada para dar soporte a la interacción usuario-sistema son las siguientes:

- ⇒ Mayor responsabilidad en las respuestas del sistema, que se fundamenta en su capacidad para conocer sus propias limitaciones y posibilidades, y que está soportada por el Módulo de Razonamiento Metanivel.
- ⇒ Permitir transparencia y accesibilidad del modelo de resolución de problemas del sistema, basadas en las características del Espacio de Resolvedores de Problemas.
- ⇒ Proporcionar mayor versatilidad en las respuestas del sistema, debido a que se apoya en la capacidad para seleccionar distintas formas de generar las respuestas en función de las posibilidades que ofrece el Espacio de Resolvedores de Problemas y del conocimiento sobre el modo de interaccionar con los distintos usuarios contenido en el modelo de Escenarios de Interacción.



- ⇒ Ofrecer mayor flexibilidad y agilidad de los diálogos que pueden mantenerse, dado que está soportada por la presencia de una Memoria de la Interacción gestionada por un mecanismo ATMS.

## 10.2 Limitaciones

En la arquitectura propuesta se han tomado una serie de decisiones de diseño que introducen limitaciones sobre sus posibles aplicaciones. Estas pueden caracterizarse como sigue:

- Actualmente, el tipo de sistemas contruidos sobre esta arquitectura tiene un alto coste de desarrollo, que se justifica en dominios problema en los que se dispone de un amplio abanico de métodos de resolución de problemas y escenarios de operación. Sin embargo, se entiende que en un futuro próximo este coste podría verse disminuido significativamente con la disponibilidad de módulos preprogramados reutilizables de resolución de problemas. Por otra parte, si se considera la gran inversión en infraestructura de comunicaciones necesaria para dar soporte a los sistemas de ayuda a la decisión actuales, el coste de incorporar en esta infraestructura un sistema desarrollado con la arquitectura ROHCI no resulta comparativamente tan elevado.
- Esta arquitectura está soportada por una memoria de la interacción a corto plazo, asociada a la sesión que se esté desarrollando con un usuario en un momento determinado, que no permite recordar características de ese usuario para futuras sesiones. Esto provoca que cada vez que un usuario inicia una nueva sesión con el sistema éste se comporte exactamente de la misma manera.
- Esta arquitectura no es especialmente adecuada en otro tipo de dominios problema especialmente interesados en los aspectos de interacción con los usuarios, como el de los sistemas tutoriales, por estar orientada a la resolución de problemas complejos en tiempo real. En el caso de un dominio como el de los sistemas tutoriales inteligentes, el modelo conceptual de interacción presentado seguiría siendo válido pero los aspectos de modelización de usuarios tendrían que ser significativamente ampliados.

## 10.3 Trabajos Futuros

A raíz de los trabajos realizados en esta tesis y de las limitaciones observadas, se abren nuevas líneas de investigación como evolución de la misma, entre las que cabe destacar las dos siguientes:

- Extender el análisis sobre estrategias comunicativas adecuadas para diferentes escenarios de interacción en distintos dominios, identificando similitudes y diferencias, que pudiesen dar lugar en el futuro a la definición de una teoría sobre estrategias de interacción hombre-máquina adaptadas a clases de dominios problema.
- Incorporar en la arquitectura ROHCI la capacidad de aprender sobre formas de interacción usuario-sistema. Como fruto de la experiencia en el uso de un sistema de estas características, podrían llegar a identificarse patrones comunicativos adecuados para determinados escenarios de interacción desarrollados por diferentes tipos de usuarios. El uso de estos patrones potenciaría la capacidad de adaptación de estos sistemas a las necesidades comunicativas de los usuarios y mejoraría la eficiencia de la interacción. Para ello habría que dotar a estos sistemas de: (i) una memoria a largo plazo en la que pudiese residir información individualizada de los usuarios, y (ii) mecanismos de aprendizaje con los que identificar y asimilar estos patrones comunicativos.





# 11. Referencias

---

- [Abascal, 96] Abascal J.: Perspectivas de la Interacción Persona-Computador. *Panorama Informático*, editado por la Federación Española de Sociedades de Informática (FESI), pag. 275-291, 1996.
- [Allen, 87] Allen J.F.: Natural Language Understanding. Menlo Park, Benjamin/Cummings Publishing Company, 1987.
- [Alonso et al., 90] Alonso M., Cuenca J., Molina M.: SIRAH: An Architecture for a Professional Intelligence. *Proceedings of the 9th European Conference on Artificial Intelligence, ECAI'90*, 1990.
- [Anderson et al., 85] Anderson J.R., Boyle F., Yost G.: The Geometry Tutor. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1, pag. 1-7, Los Angeles, CA, August 1985.
- [André et al., 93] André E., Finkler W., Graf W., Rist T., Schauder A., Wahlster W.: WIP: The Automatic Synthesis of Multimodal Presentations. *Intelligent Multimedia Presentation*, Mark T. Maybury (ed.), MIT Press, 1993.
- [Angele et al., 92] Angele J., Fensel D., Landes D., Neubert S., Studer R.: Model-Based and Incremental Knowledge Engineering: The MIKE Approach. *Proceedings of Artificial Intelligence from the Information Processing, Workshop AIFIPP'92*, Madrid, 1992.
- [Benjamins, 93] Benjamins R.: Problem Solving Methods for Diagnosis. Tesis doctoral, Universidad de Amsterdam, 1993.
- [Bennet, 85] Bennet J.S.: ROGET: A Knowledge-Based System for Acquiring the Conceptual Structure of a Diagnosis Expert System. *Journal of Automatic Reasoning*, 1, 1985.



- [Bordegoni et al., 98] Bordegoni M., Faconti G., Feiner S., Maybury M., Rist T., Ruggieri S., Trahanias P., Wilson M.: A Standard Reference Model for Intelligent Multimedia Presentation Systems. *International Journal on the Development and Applications of Standards for Computers, Data Communications and Interfaces*, vol. 18, Nos. 6 y 7, pag. 477-496, Elsevier Science, 1998.
- [Brown, Burton, 78] Brown J.S., Burton R.R.: Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. *Cognitive Science*, 2, pag. 155-192, 1978.
- [Brown, Chandrasekaran, 86] Brown D.C., Chandrasekaran B.: Knowledge and Control for a Mechanical Design Expert System. *IEEE Computer*, 19:92-101, July 1986.
- [Brown, Chandrasekaran, 89] Brown D.C., Chandrasekaran B.: Design Problem Solving. Knowledge Structures and Control Strategies. *Research Notes on Artificial Intelligence*, Pitman, Morgan Kaufmann, 1989.
- [Buchanan, Shortliffe, 84] Buchanan B., Shortliffe E.: Rule-based Expert Systems. The MYCIN Experiments. Addison Wesley, 1984.
- [Carberry, 89] Carberry S.: Plan Recognition and its Use in Understanding Dialogue. *User Models in Dialog Systems*, Kobsa A. y Wahlster W. (eds.), Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 1989.
- [Carr, Goldstein, 77] Carr B., Goldstein, I.: Overlays: A Theory of Modeling for Computer Aided Instruction. *Technical Report AI Memo 406*, AI Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1977.
- [Cawsey, 89] Cawsey A.: Explanatory Dialogues. *Interacting with Computers*, 1, pag. 69-92, 1989.
- [Chandrasekaran, 83] Chandrasekaran B.: Towards a Taxonomy of Problem Solving Types. *Artificial Intelligence Magazine* 4 (1), pag. 9-17, 1983.

- [Chandrasekaran, 86] Chandrasekaran B.: Generic Tasks in Knowledge Based Reasoning: High Level Building Blocks for Expert Systems Design. *IEEE Expert*, 1986.
- [Chandrasekaran, 87] Chandrasekaran B.: Towards a Functional Architecture for Intelligence Based on Generic Information Processing Tasks. *Proceedings of the 10th International Joint Conference on Artificial Intelligence, IJCAI'87*, pag. 1183-1192, Morgan-Kaufmann Publishers, 1987.
- [Chandrasekaran, 90] Chandrasekaran B.: Design Problem Solving: A Task Analysis. *Artificial Intelligence Magazine*, pag. 59-71, Winter 1990.
- [Chandrasekaran, Mittal, 83] Chandrasekaran B., Mittal S.: Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems. *Advances in Computers*, M. Yovits (ed.), Academic Press, pag. 217-293, 1983.
- [Chandrasekaran et al., 91] Chandrasekaran B., Swartout W.: Explanations in Knowledge Systems: The Role of Explicit Representation of Design Knowledge. *IEEE Expert*, Vol. 6, No.3, June 91.
- [Chandrasekaran et al., 92] Chandrasekaran B., Johnson T.R., Smith J.W.: Task-Structure Analysis for Knowledge Modeling. *Communications of the ACM*, vol. 35, nº 9, 1992.
- [Chin, 89] Chin D.: KNOME: Modeling What the User Knows in UC. *User Models in Dialog Systems*, Kobsa A. y Wahlster W. (eds.), pag. 74-107, Berlin: Springer-Verlag, 1989.
- [Clancey, 83] Clancey W.J.: The Epistemology of a Rule-Based Expert System - A Framework for Explanation. *Artificial Intelligence*, pag. 215-251, Mayo 1983.
- [Cohen, 78] Cohen P.R.: On Knowing What To Say: Planning Speech Acts. *TR-118*, Department of Computer Science, University of Toronto, Canada, 1978.
- [Cuenca, 89] Cuenca J.: Knowledge-Based Systems for Aid in Decision-Making: Methodology and Examples. *Perspectives in*



- Artificial Intelligence*, vol. 1, pag. 73-92, J.A. Campbell y J. Cuenca (eds.), Ellis Horwood, 1989.
- [Cuenca, 93a] Cuenca J.: Knowledge Architectures for Real Time Decision Support. *Second Generation Expert Systems*, J.M. David, J.P. Krivine y R. Simmons (eds.), pag. 689-720. Springer-Verlag, 1993.
- [Cuenca, 93b] Cuenca J.: Contributions to a Knowledge Oriented View of Software Design. J. Cuenca (ed.), Elsevier, 1993.
- [Cuenca, 97] Cuenca J.: Sistemas Inteligentes. Conceptos, Técnicas y Métodos de Construcción. Fundación General Universidad Politécnica de Madrid, ISBN 84-85632-5-0, 1997.
- [Cuenca, Hernández, 97] Cuenca J., Hernández J.: An Exercise of Knowledge Oriented Design: Architecture for Real Time Decision Support Systems. *Knowledge Based Systems: Advanced Concepts, Techniques & Applications*, Spyros G. Tzafestas (ed.), pag. 497-524, World Scientific Publishing Company, 1997.
- [Cuenca, Molina, 94] Cuenca J., Molina M.: KSM: An Environment for Knowledge Oriented Design of Applications Using Structured Knowledge Architectures. *Applications and Impacts. Proceedings of the 13th IFIP World Computer Congress, IFIP'94*, vol. 2, K. Brunnstein y E. Raubold (eds.), Elsevier Science B.V., (North-Holland), 1994.
- [Cuenca, Molina, 96] Cuenca J., Molina M.: KSM: Un Entorno para Construcción y Reutilización de Modelos de Conocimiento. *Actas del V Congreso Iberoamericano de Inteligencia Artificial, IBERAMIA'96*, pag. 215-224, Editorial Limusa, 1996.
- [Cuenca, Molina, 97] Cuenca J., Molina M.: KSM: An Environment for Design of Structured Models. *Knowledge Based Systems. Advanced Concepts, Techniques & Applications*, S.G. Tzafestas (ed.), World Scientific Publishing Company, 1997.
- [Cuenca et al., 89] Cuenca J., Molina M., Alonso M.: CYRAH: Un Ejercicio e Integración de Modelos de Simulación y Reglas para Representación del Comportamiento Físico. *Actas de las III*

- Reunión Técnica de la Asociación Española para la Inteligencia Artificial, AEPIA'89, Madrid, 1989.*
- [Cuenca et al., 94] Cuenca J., Hernández J., Molina M.: Case Presentation of the Use of Knowledge Based Models for Traffic Management - Madrid. *Proceedings of the First World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems*, ERTICO, París, 1994.
- [Cuenca et al., 95] Cuenca J., Hernández J., Molina M.: Knowledge-Based Models for Adaptive Traffic Management Systems. *Transportation Research, Part C, Issue 3 (5)*, 1995.
- [Cuenca et al., 96a] Cuenca J., Hernández J., Molina M.: Knowledge Oriented Design of an Application for Real Time Traffic Management: The TRYS System. *Proceedings of the 12th European Conference on Artificial Intelligence, ECAI'96*, W. Wahlster (ed.), Wiley, 1996.
- [Cuenca et al., 96b] Cuenca J., Hernández J., Molina M.: An Intelligent Model of Road Traffic Management in the Motorway Network Around Barcelona. *Advanced IT Tools, Proceedings of the 14th IFIP World Computer Congress, IFIP'96*, Canberra, 1996.
- [Cuenca et al., 98] Cuenca J., Hernández J., Molina M.: Advanced User Interfaces for Decision Support in Real Time Transport Management. *Fifth International Conference on Applications of Advanced Technologies in Transport Engineering, ASCE'98*, Chris T. Hendrickson, Stephen G. Ritchie (eds.), pag. 67-76, American Society of Civil Engineers, 1998.
- [Dalal et al., 96] Dalal M., Feiner S., McKeown K., Pan S., Zhou M., Höllerer T., Shaw J., Feng Y., Fromer J.: Negotiation for Automated Generation of Temporal Multimedia Presentations. *Proceedings of ACM Multimedia'96*, pag. 55-64, Association for Computing Machinery, 1996.
- [Daly-Jones et al., 97] Daly-Jones O., Bevan N., Thomas C.: Handbook of User-Centred Design. Technical Report D6.2.1, Telematics Applications Project IE 2016 (INUSE), 1997.



- [De Kleer, 86] De Kleer, J.: An Assumption-Based TMS. *Artificial Intelligence* 28, 1986.
- [De Rosis et al., 92] De Rosis F., Pizzutilo S., Russo A., Berry D.C., Nicolau J.: Modeling the User Knowledge by Belief Networks. *User Modeling and User Adapted Interaction*, 2, pag. 367-388, Kluwer Academic Publishers, Netherlands, 1992.
- [Dix et al., 93] Dix A., Finlay J., Abowd G., Beale R.: Human-Computer Interaction. Prentice-Hall, 1993.
- [Dix et al., 98] Dix A., Finlay J., Abowd G., Beale R.: Human-Computer Interaction. Second edition. Prentice-Hall, 1998.
- [Eriksson et al., 95] Eriksson H., Shahar Y., Tu S.W., Puerta A.R., Musen M.A.: Task Modeling with Reusable Problem-Solving Methods. *Artificial Intelligence* 79(2), pag. 293-326, 1995.
- [Eshelman et al., 87] Eshelman L., Eiret D., McDermott J., Tan M.: MOLE: A Tenacious Knowledge Acquisition Tool. *International Journal of Man-Machine Studies*, n° 26, 1987.
- [Feiner, McKeown, 93] Feiner S.K., McKeown K.R.: Automating the Generation of Coordinated Multimedia Presentations. *Intelligent Multimedia Interfaces*, (ed.) Mark T. Maybury, MIT Press, 1993.
- [Fensel, Harmelen, 93] Fensel D., Harmelen van F.: A Comparison of Languages which Operationalize and Formalize KADS Models of Expertise. *Research Report n° 280*, Universidad de Karlsruhe, 1993.
- [Fensel et al., 91] Fensel D., Angele J., Landes D.: KARL: A Knowledge Acquisition and Representation Language. *Proceedings of Expert Systems and Their Applications*, 1991.
- [Fink et al., 97] Fink J., Kobsa A., Schreck J.: Personalized Hypermedia Information through Adaptive and Adaptable System Features: User Modeling, Privacy and Security Issues. *Intelligence in Services and Networks: Technology for Cooperative Competition*, Mullery A., Besson M., Campolargo M., Gobbi M. and Reed R. (eds.), pag. 459-467, Springer-Verlag, 1997.

- [FLUIDS, 96] FLUIDS: Future Lines of User Interfaces for Decision Support Systems (TE2006). Telematics Applications Programme, IV Framework Programme, Commission of the European Communities. FLUIDS Consortium: Universidad Politécnica de Madrid, DFKI German Research Center for Artificial Intelligence GmbH, MIZAR Automazione S.p.A., Consorzio 5T. URL: <http://www.dfki.de/fluids>.
- [Foley et al., 88] Foley J., Gibbs C., Kim W.C., Kovacevic S. A.: A Knowledge-Based User Interface Management System. *Proceedings of the 1988 Conference on Human Factors in Computer Systems, CHI'88*, pag. 67-72, Association of Computing Machinery, 1988.
- [Foley et al., 91] Foley J., Kim W.C., Kovacevic S., Murray K.: UIDE - An Intelligent User Interface Design Environment. *Intelligent User Interfaces*, J.W. Sullivan, S.W. Tyler (eds.), pag. 339-384, ACM Press, 1991.
- [Frank, Szekely, 98] Frank M., Szekely P.: Adaptive Forms: An Interactive Paradigm for Entering Structured Data. *Proceedings of the 1998 International Conference on Intelligent User Interfaces*, pag. 153-160, 1998.
- [Garijo et al., 87] Garijo F., Verdejo F., Díaz A., Fernández Y., Sarasola K.: CAPRA: An Intelligent System to Teach Novice Programmers. In *Perspectives in Artificial Intelligence*, vol. 2, pag. 179-196, P. Campbell y J. Cuenca (eds.), Ellis Horwood, 1987.
- [Gould, Lewis] Gould J.D., Lewis C.: Designing for Usability: Key Principles and What Designer Thinks. *Communications of the ACM*, Association for Computing Machinery.
- [Grice, 75] Grice H.P.: Logic and Conversation. *Syntax and Semantics, Volume 3: Speech Acts*, P. Cole & J.L. Morgan (eds.), Academic Press, 1975.
- [Hernández, Molina, 98] Hernández J., Molina M.: Advanced Human-Computer Interaction for Decision Support Systems Using Knowledge Modeling Techniques. *Proceedings of the 15th IFIP World Computer Congress, IT&KNOWS - Information*



- Technologies and Knowledge Systems, IFIP'98, Viena-Budapest, Septiembre 1998.*
- [Hewett et al., 92] Hewett T., Baecker R., Card S., Carey T., Gasen J., Mantei M., Perlman G., Strong G. and Verplank W.. *ACM SIGCHI Curricula for Human-Computer Interaction*, Report of the ACM SIGCHI Curriculum Development Group.
- [Hix, 90] Hix D.: Generations of User-Interface Management Systems. *IEEE Software*, 7, 5, 1990.
- [Hovy, 88] Hovy E.: Planning Coherent Multisentential Text. *Proceedings of the 26th Annual Meeting of the Association of Computational Linguistics*, pag. 167-189, Buffalo, NY, 1988.
- [Hovy, 91] Hovy E.: Approaches to the Planning of Coherent Text. *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Cécile L. Paris, William R. Swartout y William C. Mann (eds.), pag. 83-102, Kluwer Academic Publishers, 1988.
- [Huang et al., 91] Huang X., Mecalla G.I., Greer J.E., Neufeld E.: Revising Deductive Knowledge and Stereotypical Knowledge in a Student Model. *User Modeling and User Adapted Interaction* 1(1), pag. 97-115, 1991.
- [Jameson et al., 80] Jameson A., Hoepfner W., Wahlster W.: The Natural Language System HAM-RPM as a Hotel Manager: Some Representational Prerequisites. R. Wilhelm (ed.), GI-10, Jahrestagung Saarbrücken, Berlin, Springer, 1980.
- [Jameson, 83] Jameson A.: Impression Monitoring in Evaluation-Oriented Dialog: The Role of the Listener's Assumed Expectations and Values in the Generation of Informative Statements. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Alemania, pag. 616-620, 1983.
- [Johnson, Soloway, 84] Johnson W.L., Soloway E.: Intention-Based Diagnosis of Programming Errors. *Proceedings of the National Conference on Artificial Intelligence*, pag. 162-168, Austin, TX, August 1984.

- [Jones, Mitchell, 95] Jones P.M., Mitchell C.M.: Human-Computer Cooperative Problem Solving: Theory, Design and Evaluation of an Intelligent Associate System. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, N° 7, 1995.
- [Josephson et al., 87] Josephson J.R., Chandrasekaran B., Smith J.W., Tanner M.C.: A Mechanism for Forming Composite Explanatory Hypothesis. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-17(3): 445-454, May/June 1987.
- [Karbach, Voß, 93] Karbach W., Voß A.: MODEL-K for Prototyping and Strategic Reasoning at the Knowledge Level. *Second Generation Expert Systems*, pag. 721-745, J.M. David, J.P. Krivine y R. Simmons (eds.) , Springer-Verlag, 1993.
- [Kass, 87] Kass R.: Implicit Acquisition of User Models in Cooperative Advisory Systems. *Technical Report MC-CIS-87-05*, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1987.
- [Kass, 89] Kass R.: Student Modelling in Intelligent Tutoring Systems - Implications for User Modelling. *User Models in Dialog Systems*, Kobsa A. y Wahlster W. (eds.), Berlin: Springer-Verlag, 1989.
- [Kass, Finin, 86] Kass R., Finin T.: User Models in Intelligent Interfaces: Why They are Needed, Problems with Their Implementation. *AAAI-86 Workshop on Intelligent Interfaces*, B. Neches y T. Kaczmarek (eds.), pag. 48-50, 1986.
- [Kay, Black, 85] Kay D., Black J.B.: The Evolution of Knowledge Representations with Increasing Expertise in Using Systems. *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, pag. 140-149, Irvine, CA, August 1985.
- [Kobsa, 90] Kobsa A.: Modeling the User's Conceptual Knowledge in BGP-MS, A User Modeling Shell System. *Computational Intelligence* 6, pag. 193-208, 1990.



- [Kobsa, Pohl, 95] Kobsa A., Pohl W.: The BGP-MS User Modeling System. *User Modeling and User-Adapted Interaction*, 4(2), pag. 59-106, 1995.
- [Kobsa et al., 94] Kobsa A., Müller D., Nill A.: KN-AHS: An Adaptive Hypertext Client of the User Modeling System BGP-MS. *Proceedings of the 4th International Conference on User Modeling*, pag. 99-105, User Modeling Inc., 1994.
- [Linster, 93] Linster M.: Using OMOS to Represent KADS Conceptual Models. *KADS - A Principled Approach to Knowledge-Based Systems Development*, pag. 221-246, G. Schreiber, B. Wielinga y J. Breuker (eds.), Academic Press, 1993.
- [Litman, Allen, 84] Litman D.J., Allen J.F.: A Plan Recognition Model for Clarification Subdialogues. *Proceedings of the 10th International Conference on Computational Linguistics*, Palo Alto, CA, July 1984.
- [Macintyre, 93] Macintyre A.: KREST User Manual 2.5. Wrije Universiteit Brussels, AI Lab, 1993.
- [Macmillan, 83] Macmillan S.A.: User Models to Personalize an Intelligent Agent. Tesis Doctoral, School of Education, Stanford University, CA, 1983.
- [Mann, Thompson, 87] Mann W., Thompson S.: Rethorical Structure Theory: Description and Construction of Text Structures. *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, G. Kempen (ed.), NATO ASI Series E 135, pag. 85-95, 1987.
- [Marcus, McDermott, 89] Marcus S., McDermott J.: SALT: A Knowledge Acquisition Tool for Propose-and-Revise Systems. *Artificial Intelligence*, vol. 39, 1989.
- [Matthews, Biswas, 86] Matthews M., Biswas G.: USCSH: An Active Assistance Interface for Unix. University of South Carolina, *Technical Report USC-CS TR 86-003*, June 12, 1986.

- [Maybury, 92] Maybury M.T.: Communicative Acts for Explanation Generation. *International Journal of Man-Machine Studies*, Academic Press, 1992.
- [Maybury, 93] Maybury M.T.: Planning Multimedia Explanations using Communicative Acts. *Intelligent Multimedia Interfaces*, M.T. Maybury (ed.), pag. 59-74, MIT Press, 1993.
- [Maybury et al., 97] Maybury M.T., Merlino A., Morey D.: Broadcast News Navigator Using Story Segments. *ACM International Multimedia Conference*, Seattle, pag. 381-391, ACM Press, 1997.
- [McDermott, 88] McDermott J.: Preliminary Steps Towards a Taxonomy of Problem-Solving Methods. *Automating Knowledge Acquisition for Expert Systems*, Marcus S. (ed.), pag. 225-256, Kluwer Academic Publishers, 1988.
- [McGraw, 92] McGraw K.L.: Designing and Evaluating User Interfaces for Knowledge-Based Systems. Series in Interactive Information Systems, Ellis Horwood, 1992.
- [McKeown, 82] McKeown K.R.: Generating Natural Language Text in Response to Questions at Database Structure. Tesis Doctoral, Universidad de Pensilvania, TR MIS-CIS-82, 1982.
- [McKeown, 85] McKeown K.: Text Generation. Cambridge, UK. Cambridge University Press, 1985.
- [McKeown et al., 90] Mc Keown K., Elhadad M., Fukumoto Y., Lim J., Lombardi C., Smadja F.: Language Generation in COMET. *Current Research in Natural Language Generation*, Mellish C., Dale R., Zock M. (eds.), pag. 103-140, Academic Press, 1990.
- [Mittal, Moore, 95] Mittal V.O., Moore J.D.: Dynamic Generation of Follow-up Question Menus: Facilitating Interactive Natural Language Dialogues. *Proceedings of the Human Factors in Computing Systems Conference, CHI'95*, pag. 90-97, 1995.
- [Molina, 93] Molina M.: Desarrollo de Aplicaciones a Nivel Cognitivo Mediante Entornos de Conocimiento Estructurado. Tesis



- Doctoral, Facultad de Informática de la Universidad Politécnica de Madrid, 1993.
- [Molina, Cuenca, 95] Molina M., Cuenca J.: Knowledge Oriented and Object Oriented Design: The Experience of KSM. *Proceedings of the Knowledge Acquisition Workshop, KAW'95, Banff, 1995.*
- [Molina et al., 98a] Molina M., Sierra J.L., Serrano J.M.: A Language to Formalize and to Operationalize Problem Solving Strategies of Structured Knowledge Models. *Proceedings of the 8th Workshop on Knowledge Engineering, Methods & Languages, KEML'98, Karlsruhe, 1998.*
- [Molina et al., 98b] Molina M., Hernández J., Cuenca J.: A Structure of Problem-Solving Methods for Real-Time Decision Support in Traffic Control. *International Journal on Human and Computer Studies* 49, 1998.
- [Moore, 89] Moore J.D.: A Reactive Approach to Explanation in Expert and Advice-Giving Systems. Tesis Doctoral. Universidad de California en Los Angeles, 1989.
- [Moore, 95] Moore J.D.: Participating in Explanatory Dialogues. Interpreting and Responding to Questions in Context. MIT Press, 1995.
- [Moore, Paris, 93] Moore J.D., Paris C.L.: Planning Text for Advisory Dialogues: Capturing Intentional and Rethorical Information. *Computational Linguistics*, 19(4), pag. 651-695, Association of Computational Linguistics, 1993.
- [Morik, 89] Morik K.: User Models and Conversational Settings: Modeling the User's Wants. *User Models in Dialog Systems*, Kobsa A. y Wahlster W. (eds.), Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 1989.
- [Morik, Rollinger, 85] Morik K., Rollinger C.R.: The Real Estate Agent - Modeling the User by Uncertain Reasoning. *AI Magazine*, 6 (2), pag. 44-52, Summer 1985.
- [Musen, 89] Musen M.: Automated Support for Building and Extending Expert Models. *Machine Learning*, 4, 1989.

- [Musen et al., 95] Musen M.A., Gennari J.H., Eriksson H., Tu S.W., Puerta A.R.: PROTÉGÉ-II: Computer Support for Development of Intelligent Systems from Libraries of Components. *Proceedings of MEDINFO'95, 8th World Congress on Medical Informatics*, pag. 766-770, 1995.
- [Neches et al. 85] Neches R., Swartout W.R., Moore J.D.: Enhanced Maintenance and Explanation of Expert Systems Through Explicit Models of Their Development. *IEEE Transactions on Software Engineering*, vol. SE-11, N° 11, Noviembre 1985, pag. 1337-1351.
- [Newell, 82] Newell A.: The Knowledge Level. *Artificial Intelligence* 18, pag. 87-127, 1982.
- [Norcio, Stanley, 89] Norcio A., Stanley J.: Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. *IEEE Transactions on Systems, Man and Cybernetics* 19, 399-408, 1989.
- [Pohl, Höhle, 97] Pohl W., Höhle J.: Mechanisms for Flexible Representation and Use of Knowledge in User Modeling Shell Systems. *User Modeling: Proceedings of the 6th International Conference*, Jameson A., Paris C. y Tasso C. (eds.), pag. 403-414, Springer, URL: <http://zeus.gmd.de/hci/projects/bgp-ms>, 1997.
- [Puerta, 97] Puerta A.: Model-Based Development Tools. *IEEE Software*, 14(4), pag. 40-47, 1997.
- [Puerta, Maulsby, 97] Puerta A., Maulsby D.: MOBI-D: A Model-Based Development Environment for User-Centered Design. *Proceedings of CHI'97 (Extended abstracts)*, 1997.
- [Puerta et al., 93] Puerta A., Tu S.W., Musen M.: Modeling Tasks with Mechanisms. *International Journal of Intelligent Systems*, vol. 8(1), pag. 129-152, John Wiley & Sons, 1993.
- [Puerta et al., 94] Puerta A.R., Eriksson H., Gennari J.H., Musen M.: Model-Based Automated Generation of User Interfaces. *Proceedings*



- of the National Conference on Artificial Intelligence*, pag. 471-477, American Association for Artificial Intelligence, 1994.
- [Puerta et al., 95] Puerta A.R., Eriksson H., Gennari J.H., Musen M.: Beyond Data Models for Automated User Interface Generation. 1995.
- [Punch, Chandrasekaran, 93] Punch W.F., Chandrasekaran B.: An Investigation of the Roles of Problem-Solving Methods in Diagnosis. *Second Generation Expert Systems*, J.M. David, J.P. Krivine y R. Simmons (eds.), pag. 673-688. Springer-Verlag, 1993.
- [Reiser et al., 85] Reiser B., Anderson J.R., Farrel R.G.: Dynamic Student Modeling in an Intelligent Tutor for Lisp Programming. *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, 1, pag. 8-14, Los Angeles, CA, August 1985.
- [Rich, 79] Rich E.: User Modelling via Stereotypes. *Cognitive Science* 3, pag. 329-345, 1979.
- [Rich, 83] Rich E.: Users are Individuals: Individualizing Users Models. *International Journal of Man-Machine Studies* 18, pag. 199-214, 1983.
- [Rich, 89] Rich E.: Stereotypes and User Modelling. *User Models in Dialog Systems*, Kobsa A. y Wahlster W. (eds.), pag. 35-51, Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 1989.
- [Rist et al., 97] Rist T., André E., Müller J.: Adding Animated Presentation Agents to the Interface. *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, Orlando, pag. 79-86, 1997.
- [Serrano, 96] Serrano J.M.: Modelo de Conversación de un Sistema de Ayuda a la Decisión en Situaciones de Emergencia. Proyecto Fin de Carrera, Facultad de Informática, Universidad Politécnica de Madrid, 1996.
- [Sierra, 95] Sierra J.L: Formalización e Instrumentación de la Dinámica de los Modelos Basados en Unidades Cognitivas. Proyecto

- Fin de Carrera, Facultad de Informática, Universidad Politécnica de Madrid, 1995.
- [Sierra, Molina, 98] Sierra J.L., Molina M.: Terminological Importation for Adapting Reusable Knowledge Representation Components in the KSM Environment. *Workshop on Application of Ontologies and Problem Solving Methods, 13th European Conference on Artificial Intelligence, ECAI'98*, 1998.
- [Sleeman, 85] Sleeman D.H.: UMFE: A User Modelling Front End Subsystem. *International Journal of Man-Machine Studies* 23, pag.71-88, 1985.
- [Sleeman, Brown, 82] Sleeman D.H., Brown J.S.. *Intelligent Tutoring Systems*. New York, Academic Press, 1982.
- [Sleeman, Smith, 81] Sleeman D.H., Smith M.J.: Modelling Student's Problem Solving. *Artificial Intelligence*, 16, pag. 171-187, 1981.
- [Steels, 90] Steels L.: Components of Expertise. *AI Magazine*, Summer 1990.
- [Steels ,92] Steels L.: Reusability and Configuration of Applications by Non-Programmers. *AI Memo 92-4*, AI-Lab, Vrije Universiteit Brussel, 1992.
- [Stevens et al., 79] Stevens A., Collins A., Goldin S.E.: Misconceptions in Student's Understanding. *International Journal on Man-Machine Studies*, 11, pag. 145-156, 1979.
- [Swartout, 77] Swartout W.R.: A Digitalis Therapy Advisor with Explanations. Master Thesis, MIT, TR-176, 1977.
- [Szekely et al., 92] Szekely P., Luo P., Neches R.: Facilitating the Exploration of Interface Design Alternatives: The HUMANOID Model of Interface Design. *Proceedings of CHI'92*, pag. 507-515, 1992.
- [Szekely et al., 93] Szekely P., Luo P., Neches R.: Beyond Interface Builders: Model-Based Interface Tools. *Proceedings of INTERCHI'93*, pag. 383-390, 1993.



- [Tu et al., 92] Tu S.W., Shahar Y., Davis J., Winkles J., Puerta A., Musen M.: A Problem-Solving Model for Episodic Skeletal-Plan Refinement. *Knowledge Acquisition*, 4, 1992.
- [Tu et al., 95] Tu S.W., Eriksson H., Gennari J., Shahar Y., Musen M.A.: Ontology-Based Configuration of Problem-Solving Methods and Generation of Knowledge-Acquisition Tools: Application of PROTÉGÉ-II to Protocol-Based Decision Support. *Artificial Intelligence in Medicine* 7(3), pag. 257-289, 1995.
- [Vanwelkenhuysen, Rademakers, 90] Vanwelkenhuysen J., Rademakers P.: Mapping a Knowledge Level Analysis Onto a Computational Framework. *Proceedings of the 9th European Conference on Artificial Intelligence, ECAI'90*. London, Pitman Publishing, 1990.
- [Vanwelkenhuysen, Rademakers, 93] Vanwelkenhuysen J., Rademakers P.: Generic Models and Their Support in Modeling Problem Solving Behavior. *Second Generation Expert Systems*, J.M. David, J.P. Krivine y R. Simmons (eds.), pag. 350-375. Springer-Verlag, 1993.
- [Verdejo, 92] Verdejo M.F.: User Modelling in Knowledge-Based Systems. *Cognition, Semantics and Philosophy, Proceedings of the First International Colloquium on Cognitive Science*, Jesús Ezquerro y Jesús M. Larrazabal (eds.), pag. 23-45, Kluwer Academic Publishers, 1992.
- [Verdejo, 93] Verdejo M.F.: Building a Student Model for an Intelligent Tutoring System. *Student Modeling: The Key to Individualized Knowledge-Based Instruction*, Jim E. Greer y Gordon Y. McCalla (eds.), pag. 147-164, Springer-Verlag, 1993.
- [Wahlster, 88] Wahlster W.: Distinguishing User Models from Discourse Models. *Computational Linguistics*, Koban A. y Wahlster W. (eds.), 14(3), pag. 101-103, 1988.
- [Wahlster, 93] Wahlster W.: User Modeling and Multimodal Discourse. Course notes for the Advanced Course on Artificial Intelligence, ACAI'93, 1993.

- 
- [Wahlster et al., 93] Wahlster W., André E., Finkler W., Profitlich J., Rist T.: Plan-Based Integration of Natural Language and Graphics Generation. *Artificial Intelligence*, 63(1-2), pag. 387-428, 1993.
- [Wahlster, Kobsa, 86] Wahlster W., Kobsa A.: Dialog-based User Models. *Proceedings of IEEE*, 74(7), 1986.
- [Wielinga et al., 92] Wielinga B. J., Schreiber A.T., Breuker J.A.: KADS: A Modeling Approach to Knowledge Engineering. *Knowledge Acquisition*, 4(1), 1992.
- [Wilson et al., 93] Wilson S., Johnson P., Markopoulos P., Pycock J.: ADEPT - Advanced Design Environment for Prototyping with Task Models. *Proceedings of INTERCHI'93*, ACM Press, 1993.
- [Wilson et al., 94] Wilson S., Johnson P., Kelly C., Cunningham J., Markopoulos P.: Beyond Hacking: A Model-Based Approach to User Interface Design. 1994.





**ANEXO**





# Gestión del Diálogo Multimedia en el Proyecto FLUIDS

---

La validación de la arquitectura ROHCI se ha llevado a cabo en el marco del proyecto FLUIDS [FLUIDS, 96]. El complemento necesario a la organización de conocimiento propuesta en esta tesis es el interfaz de usuario a través del cual se canaliza la información generada por el modelo de resolución de problemas y se recogen las consultas del usuario. En el caso del proyecto FLUIDS este elemento es especialmente relevante al tratarse de un generador de presentaciones inteligentes multimedia basado en WIP (ver apartado 3.3.2); del que en este anexo se incluye una breve descripción.

## 1. Introducción

El componente multimedia del entorno FLUIDS es responsable de la interacción directa con el usuario y sirve como mediador entre el modelo de resolución de problemas y el operador humano que debe tomar las decisiones. El gestor del diálogo incluido en este componente necesita, por una parte, recoger las entradas del usuario y traducirlas en consultas adecuadas para el modelo de resolución de problemas, y por otra parte, elaborar formas de presentar la información obtenida como resultado de procesos internos de razonamiento.

Ahora bien, para poder gestionar la dinámica de la comunicación entre el usuario y el sistema de ayuda a la decisión, es necesario un nivel de control de diálogo más elevado. En este caso, el gestor del diálogo tiene que estructurar la interacción para poder exhibir un comportamiento en el diálogo que sea sensible al contexto. Los diálogos multimedia pueden considerarse como una secuencia de actos comunicativos con los que alcanzar determinados objetivos. Desde esta perspectiva, el control del diálogo estaría relacionado con el reconocimiento y planificación de actos comunicativos, y la presentación de



información se caracterizaría como una actividad guiada por objetivos que persigue la realización de un acto comunicativo complejo.

## 2. Interacción con el Modelo de Resolución de Problemas

La interacción entre el gestor del diálogo y el modelo de resolución de problemas está soportada por un protocolo de comunicación basado en predicados. Estos predicados se usan para configurar el contenido de las preguntas y respuestas. La estructura de una pregunta interpretada por el gestor del diálogo siempre incluye el objeto sobre el que se formula la pregunta, alguna condición o acción relacionada con el contexto del objeto, y los modos de interacción, descritos en el apartado 6.4.2.1, asociados a la pregunta. Una vez que el gestor del diálogo le pasa una pregunta al modelo de resolución de problemas, éste notifica la recepción del mensaje con una estimación del tiempo requerido para generar la correspondiente respuesta. Todas las respuestas generadas por el modelo de resolución de problemas tienen la misma estructura, en la que se distinguen tres partes:

- *Descripción.* Incluye una formulación a alto nivel de la información requerida sobre el objeto de la pregunta.
- *Elaboración.* Incluye una descripción de los aspectos que dan soporte a la respuesta proporcionada en el argumento *description*, en el argumento *elaboration*. Esta parte puede hacer un uso recursivo de la estructura de la respuesta si la información del objeto está soportada por información relacionada con los componentes del objeto. Por ejemplo, el estado de una línea se obtiene a partir del estado de los vehículos que circulan por ella, por lo que la clase de situación observada en una línea se incluiría en el argumento *description* mientras que el argumento *elaboration* incluiría el estado de los vehículos con problemas en la línea.
- *Explicación.* Contiene información en forma de texto que justifica las conclusiones ofrecidas en los argumentos anteriores.

Mensajes del gestor del diálogo	Mensajes del modelo de resolución de problemas
<b>start</b>  <b>whatIsHappening(OBJECT, MODE )</b> OBJECT = network(N), line(L) or vehicle(V)  MODE = lista de algunos de los siguientes predicados:  typeOfOperator(T) levelOfAbstraction(L) timeConstraints(C) degreeOfSupport(D) temporalDepth(P) ...	<b>accepted</b>  <b>acknowledge(SECONDS )</b> SECONDS = tiempo previsto de respuesta en segundos <b>currentSituation( DESCRIPTION, ELABORATION, EXPLANATION)</b> DESCRIPTION = lista de algunos de los siguientes predicados:  vehicleState(VEH, STATE) lineState(LINE, STATE) networkState(NETWORK, STATE) lineBlockedStops(LINE, STOPS) ... ELABORATION = lista de predicados del tipo currentSituation(...)  EXPLANATION = texto ascii con la explicación <b>currentSituation( DESCRIPTION, ELABORATION, EXPLANATION)</b> ... <b>endOfAnswers</b>
<b>whatMayHappen( OBJ, CONDITIONS, ACTIONS, MODE)</b> OBJ = network(N), line(L)  CONDITIONS = uno de los siguientes predicados:  currentConditions, averageSpeedOfLine(LINE, SPEED)  ACTIONS = uno de los siguientes predicados:  noActions, removeVehicle(VEHICLE)	<b>acknowledge(SECONDS )</b> <b>foreseeableSituation(DESCRIPTION, ELABORATION, EXPLANATION)</b> DESCRIPTION = lista de algunos de los siguientes predicados:  lineDirectionState(LD, STATE) lineDirectionTimeOf Departure(LD, TIME) ... ELABORATION = lista de predicados del tipo foreseeableSituation(...) <b>foreseeableSituation(DESCRIPTION, ELABORATION, EXPLANATION)</b> ... <b>endOfAnswers</b>
<b>whatToDo( OBJECT, CONDITIONS, MODE)</b> OBJECT = network(N), line(L)  CONDITIONS =uno de los siguientes predicados:  currentConditions, averageSpeedOfLine(LINE, SPEED) unavailableReserveVehicles(LINE) estimatedTimeForReparation(VEH, MINUTES) canReachEndOfLine(VEH)	<b>acknowledge(SECONDS )</b> <b>proposedAction( DESCRIPTION, ELABORATION, EXPLANATION)</b> DESCRIPTION = lista de algunos de los siguientes predicados:  limitation(VEHICLE, NEXT_VEHICLE) spola(VEHICLE, LINE, END_LINE, TIME_DEPARTURE) detour(VEHICLE, LINE, END_LINE, STOPS) .... ELABORATION = lista de predicados del tipo proposedAction(...) <b>proposedAction( DESCRIPTION, ELABORATION, EXPLANATION)</b> ... <b>endOfAnswers</b>
<b>end</b>	<b>accepted</b>

Figura 1: Protocolo de comunicación entre el gestor del diálogo y el modelo de resolución de problemas

Además, asociadas a las tres grandes clases de preguntas planteables por un usuario, pueden distinguirse tres clases de patrones pregunta-respuesta (ver figura 1):

- Una estructura *current\_situation* que proporciona las respuestas a las preguntas del tipo *what\_is\_happening(object, mode)*, donde *object* es una línea de transporte público o un vehículo concreto, y *mode* es una lista de valores para los modos de interacción accesibles al usuario.
- Una estructura del tipo *foreseeable\_situation* relacionada con la clase *what\_may\_happen(object, conditions, actions, mode)*. Los parámetros *object* y



*mode* son análogos a los de la clase anterior, y los parámetros *conditions* y *actions* se usan para expresar preguntas condicionales.

- De forma similar, la estructura *proposed\_action* se corresponde con la clase de preguntas *what\_to\_do(object, conditions, mode)*.

Los predicados utilizados en la definición de los argumentos *description* y *elaboration* de cualquier respuesta se pueden referir a (i) información de estado de vehículos, líneas, determinadas direcciones de circulación en las líneas o la red de transporte público completa, o bien a (ii) acciones de control simples o compuestas.

### 3. Interacción Multimedia Basada en Planes

En FLUIDS, el diseño de los diálogos multimedia en el resultado de un proceso de planificación basado en el conocimiento en el que se llevan a cabo tres tareas de diseño:

- diseño de la estructura de diálogo,
- diseño de estructuras de presentación con las que transmitir actos comunicativos,
- diseño de los objetos asociados a los diferentes medios empleados para presentar información

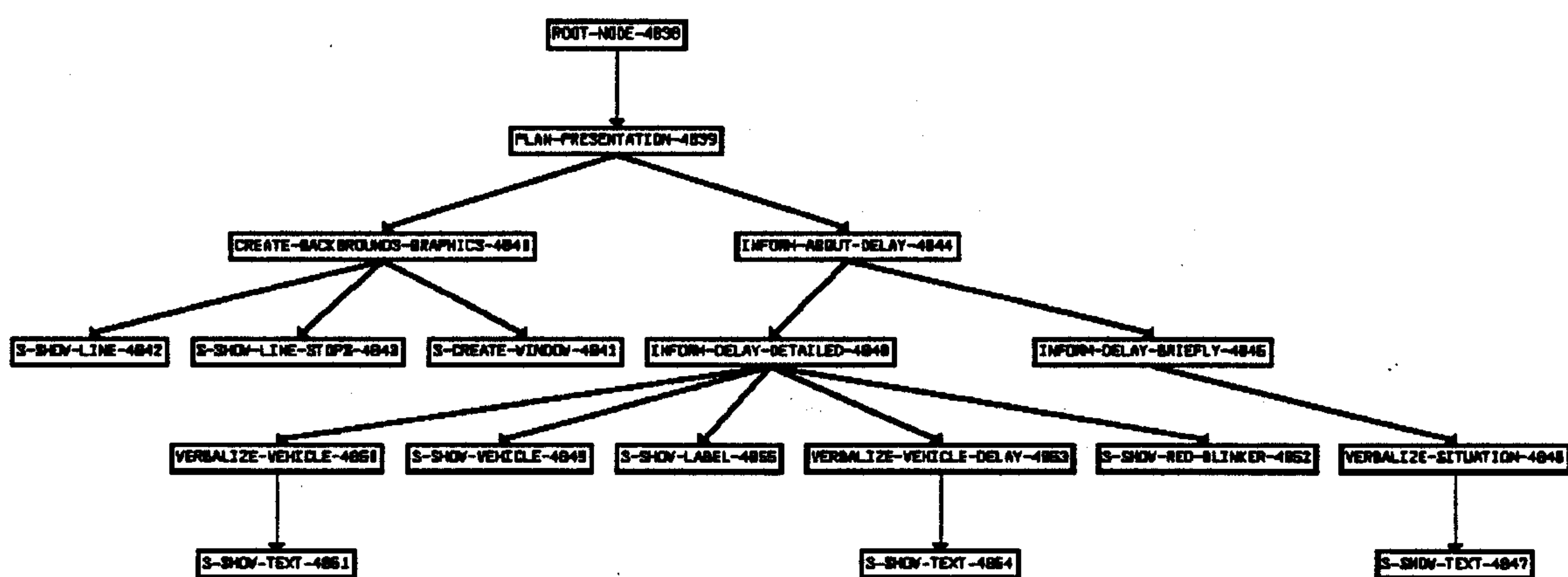


Figura 2: Plan de diseño de una presentación multimedia



A partir de aquí, el componente planificador de la presentación va configurando en cada ciclo de inferencia los diferentes pasos del diseño de la presentación organizados en un *pipeline* de tareas. El conocimiento sobre las técnicas de diseño de presentaciones está representado por estrategias de diseño declarativas que son tratadas como operadores del planificador. Tanto las estrategias de diseño como las estructuras de diseño resultantes de la planificación (ver figura 2) están basadas en conceptos teóricos de actos de habla adaptados del procesamiento del lenguaje natural a la interacción multimedia.

La figura 3 muestra algunos ejemplos de estrategias de presentación específicas para la aplicación de gestión del transporte público. Estos operadores pueden aplicarse para generar la presentación de una situación problemática en la que algunos vehículos de una línea concreta están retrasados respecto a su horario previsto. Como puede observarse en estas estrategias, la planificación de la presentación tiene que emplear razonamiento temporal para manejar las restricciones temporales cuantitativas y cualitativas de la coordinación dinámica multimedia.

```
(define-plan-operator
:HEADER (A0 (PLAN-PRESENTATION P A What-Is-Happening ?query))
:CONSTRAINTS (BEL P (?query line-status non-uniform-delay))
:INFERIORS ( (A1 (CREATE-BACKGROUNDS-GRAPHICS P A ?query))
              (A2 (INFORM-ABOUT-DELAY P A ?query)))
:QUALITATIVE ((A1 (m) A2))
:START A1
:FINISH A2)

(define-plan-operator
:HEADER (A0 (CREATE-BACKGROUNDS-GRAPHICS P A ?query))
:CONSTRAINTS ("AND" (BEL P (?query text ?text-window))
              (BEL P (?query graphic ?graphic-window)))
:INFERIORS ( (A1 (S-CREATE-WINDOW P A ?text-window ?graphic-window))
              (A2 (S-SHOW-LINE P A ?graphic-window ?line))
              (A3 (S-SHOW-LINE-STOP P A ?graphic-window ?line)))
:QUALITATIVE ((A1 (m) A2) (A2 (e) A3))
:START A1
:FINISH A3)

(define-plan-operator
:HEADER (A0 (INFORM-DELAY-DETAILED P A ?text-window
              ?graphic-window ?y-pos-1 ?y-pos-2
              ?vehicle ?v-location ?v-delay
              ?delay-label))
:INFERIORS ( (A1 (S-SHOW-VEHICLE P A ?graphic-window ?vehicle
              ?v-location ?y-pos-1))
              (A2 (VERBALIZE-VEHICLE P A ?text-window ?vehicle)))
```



```

(A3 (S-SHOW-RED-BLINKER P A ?graphic-window ?v-location
      ?y-pos-1))
(A4 (VERBALIZE-VEHICLE-DELAY P A ?text-window ?minutes))
(A5 (S-SHOW-LABEL P A ?graphic-window ?delay-label
      ?v-location ?y-pos-2)))
:QUALITATIVE ((A1 (e) A2) (A2 (s) A3) (A2 (m) A4) (A4 (m) A5))
:METRIC ((20 <= DURATION A3 <= 30))
:START A3
:FINISH A3)

```

Figura 3: Operadores de planificación para describir un problema de retrasos en una línea

## 4. Proceso de Generación de Presentaciones

A partir de la estructura de la presentación obtenida anteriormente se determina una agenda detallada de la ejecución de la presentación teniendo en cuenta las restricciones temporales impuestas por el plan de la presentación. La figura 4 muestra la agenda para la presentación del plan descrito en la figura 2.

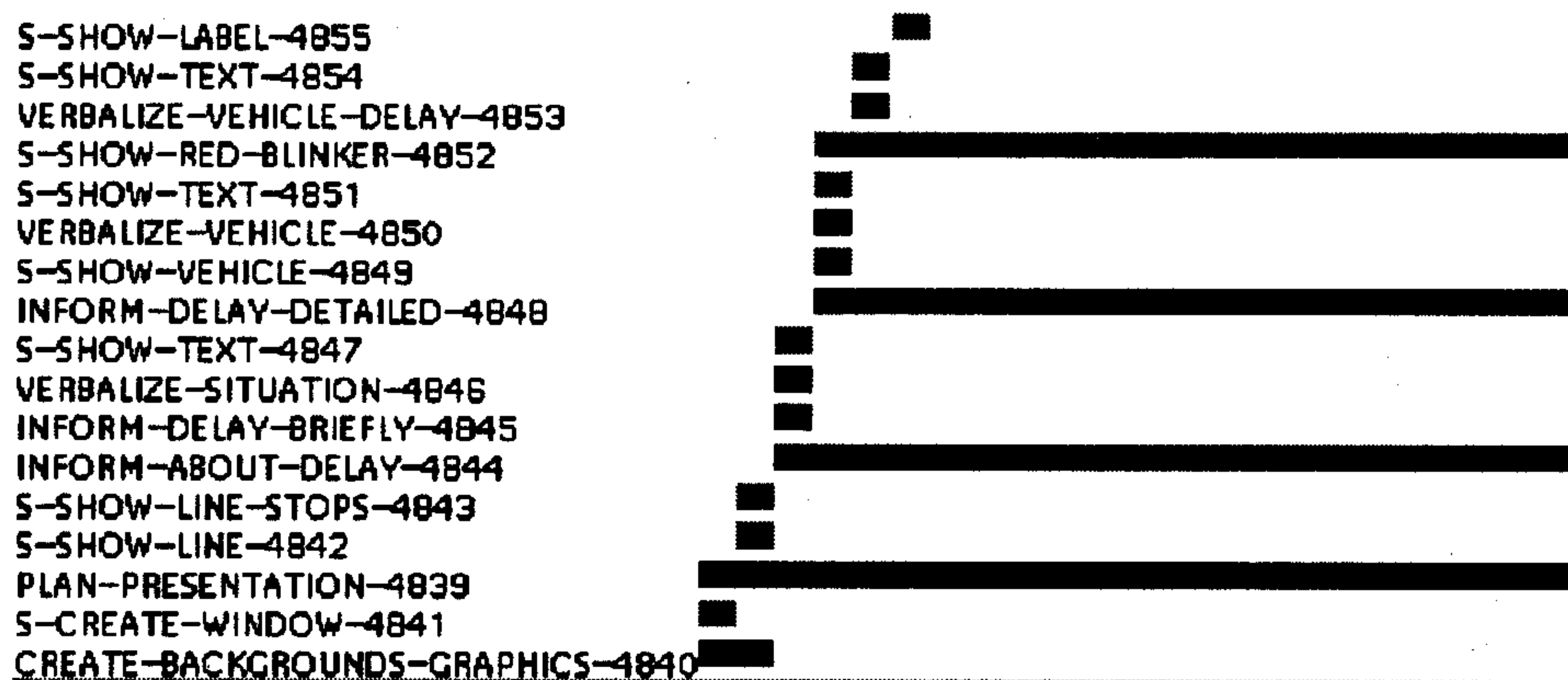


Figura 4: Agenda para ejecutar un plan de presentación

```

(SetDialogState
  (QueryWhatMayHappen ("line_direction("NEGARVILLE")"
    "nil" "nil" "What may happen"))
  (QueryWhatToDo ("line(63)" "nil" "What can be done")))

(SetLayout$VerticalTextGraphics [0 1] Text Graphics 5 20)
(ShowBusLine [2 3] Graphics 63)
(ShowAllLineStops [2 3] Graphics 63)
(ShowText [3 4] Text
  "A non-uniform delay situation has been detected for line 63.")
(ShowVehicle [5 6] Graphics 1344 18832 25)

```



```
(ShowVehicle [5 6] Graphics 1460 16832 25)
(ShowVehicle [5 6] Graphics 1397 14976 25)
(ShowVehicle [5 6] Graphics 1484 14876 25)
(ShowVehicle [5 6] Graphics 1390 10936 25)
(ShowVehicle [5 6] Graphics 1463 9728 25)
(ShowVehicle [5 6] Graphics 1399 5904 25)
(ShowVehicle [5 6] Graphics 1396 3288 25)
(ShowText [15 16] Text "Vehicle no. 1397 is affected.")
(ShowVehicle [15 16] Graphics 1397 14976 25)
(ShowRedBlinker [15 2000] Graphics 14976 25)
(ShowText [22 23] Text "It has a delay of about 12 minutes.")
(ShowLabel [23 24] Graphics "-12" 14976 45)
```

Figura 5: Script de presentación para un escenario de retrasos en una línea

El resultado final del proceso de planificación es un *script* de presentación que a su vez puede ser ejecutado para generar la salida perceptible en el interfaz. La figura 5 contiene el *script* de presentación asociado al escenario de retraso considerado en las figuras anteriores. El resultado de ejecutar este *script* es el que se muestra en la figura 6. Como puede observarse en este ejemplo, las posibles acciones que puede realizar el usuario para continuar el diálogo son controladas por el proceso generador de la presentación. En este caso, una vez notificado el problema de retrasos en la línea, el diseño del diálogo posterior considera la posibilidad de formular preguntas del tipo *what\_may\_happen* y *what\_to\_do* con las que poder obtener más información sobre la situación de esa línea o bien solicitar acciones correctivas.

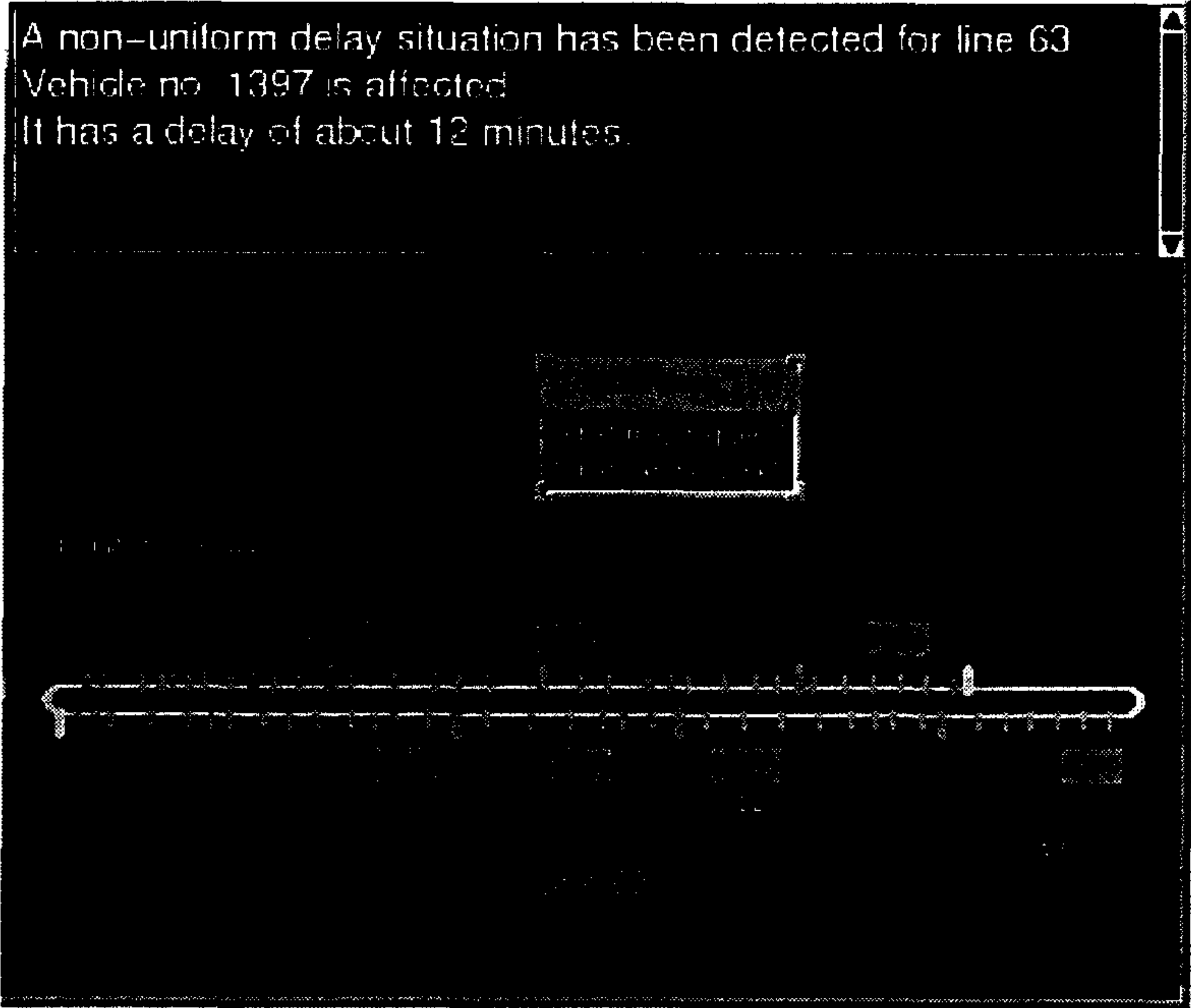


Figura 6: Presentación resultante



